A Thermal Model to Optimize Performance in Green Roofs

A Master of Engineering Project Report
Presented to the Faculty of the College of Engineering and
The Department of Biological and Environmental Engineering of Cornell University

By
William Striar Lambert
May 2006

Table of Contents

List of Figures ..... **iv**

List of Tables ..... **v**

Introduction **1**

Methods **2**
- Model Formulation 2
- Parameters Used 9
- Weather Data 10
- Program 12

Results and Discussion **16**
- Model Validation 16
- Sensitivity Analysis 20
- Temperature Profile 30
- Flux Analysis 34

Conclusion **37**

Acknowledgements **40**

References **42**

Appendix A **44**

Appendix B **46**

Appendix C **47**

Appendix D **49**

Appendix E **65**

List of Figures

Figure 1: Schematic of modeled green roof system ... 3
Figure 2: Representation of interaction between main program and supporting functions ... 13
Figure 3: Temperature variation throughout model validation for T_0-T_4 ... 18
Figure 4: Temperature variation throughout model validation for T_5 ... 18
Figure 5: Temperature variation throughout model validation for T_g ... 19
Figure 6: Temperature variation throughout model validation for T_c ... 19
Figure 7: Thermal conductivity of the structural support sensitivity analysis for the NY simulation, layer T_0 ... 22
Figure 8: Thermal conductivity of the structural support sensitivity analysis for the NY simulation, layer T_1 ... 23
Figure 9: Thermal conductivity of the structural support sensitivity analysis for the CA simulation, layer T_0 ... 24
Figure 10: Thermal conductivity of the structural support sensitivity analysis for the CA simulation, layer T_1 ... 24
Figure 11: LAI sensitivity analysis for the NY simulation, layer T_0 ... 25
Figure 12: LAI sensitivity analysis for the NY simulation, layer T_c ... 26
Figure 13: Soil depth sensitivity analysis for the NY simulation, layer T_0 ... 27
Figure 14: Soil depth sensitivity analysis for the CA simulation, layer T_0 ... 28
Figure 15: Thermal conductivity of the soil sensitivity analysis for the NY simulation, layer T_0 ... 29
Figure 16: Thermal conductivity of the soil sensitivity analysis for the CA simulation, layer T_0 ... 30
Figure 17: Temperature profile on JD 198 for NY simulation ... 31
Figure 18: Temperature profile on JD 198 for CA simulation ... 32
Figure 19: Temperature profile on JD 198 for AZ simulation ... 33
Figure 20: Schematic of the T_0 layer including FLUX OUT term ... 34
Figure 21: Flux out analysis for the NY simulation comparing a green roof to a non-green roof ... 35
Figure 22: Flux out analysis for the CA simulation comparing a green roof to a non-green roof ... 36
Figure 23: Flux out analysis for the AZ simulation comparing a green roof to a non-green roof ... 36

List of Tables

Table 1: Seasonal values for convection coefficient of air and wind speed 10
Table 2: Coefficients A_1-A_7 used in Equation 14 11
Table 3: Variables from Equation 13 and their representative function names................ 14
Table 4: Weather data used in model validation... 17
Table 5: Layer analysis for model validation 20
Table 6: Flux out analysis results for the three modeled locations.................................. 37
Table 7: Amount of energy saved by installing a green roof... 37

Introduction

A green roof is simply any roof that is partially or completely covered in plants. This alternative building technology was developed and has been used extensively in Europe, although it has begun to be researched and developed for use in the United States. There are two types of green roofs, intensive and extensive. Intensive green roofs, or rooftop gardens, are common in many locations; however they only provide an aesthetic benefit and lack a significant engineering advantage to a structure. Extensive green roofs usually contain one or two plant species and are designed to optimize thermal and hydrological performance (Wark 2003), as well as improve the surrounding air quality. Thermal benefits occur when water evaporates from the plants surface to provide cooling; the plants absorb the Sun's heat during the day and dew forms at night (Thompson 1998). For this study an extensive green roof system was assumed to model performance.

Much of the research on extensive green roof systems has been used to identify plant species other than *Sedum* that promote thermal and hydrological performance, as well as appropriate growing media depth and soil moisture content (Dunnett and Nolan 2004). In addition to this plant science approach, observations on thermal performance have been made on existing green roof structures. It has been found that extensive green roofs lower a building's cooling load by 20%-30% and reduce the indoor air temperature by as much as 4°C (Bass 2001). Other studies have found that green roofs have remained cooler and had lower median temperature fluctuations than regular roofs, while also investigating how to improve performance by varying shading, insulation, evapotranspiration, and thermal mass (Liu and Bakaran 2003).

The above previously conducted research only provides data for those specific case studies but is not a general model that can be applied to all green roofs. Currently there is a serious lack of quantifiable data and an energy model for extensive green roofs (Rowe 2002). In order to rectify this problem, an energy model and a subsequent program to calculate temperature profile's in green roofs was created. The approach, development of this model and the results are provided below.

Methods

Model Formulation

The heat transfer model was developed by solving the energy balance for each node in the model, utilizing a lumped parameter approach, with one-dimensional, time-dependent, heat transfer in the system. This reasoning was partially inspired by an article, which provided information regarding boundary conditions, and other properties important to a comprehensive green roof model (Barrio 1998). The model and approach presented was used as a guide to develop a unique model that could be run for a variety of properties and locations.

There are three sections to the modeled green roof (canopy, soil, and structural support), broken into 8 layers for the energy balance. Figure 1 below shows a schematic of a cross-section of a green roof.

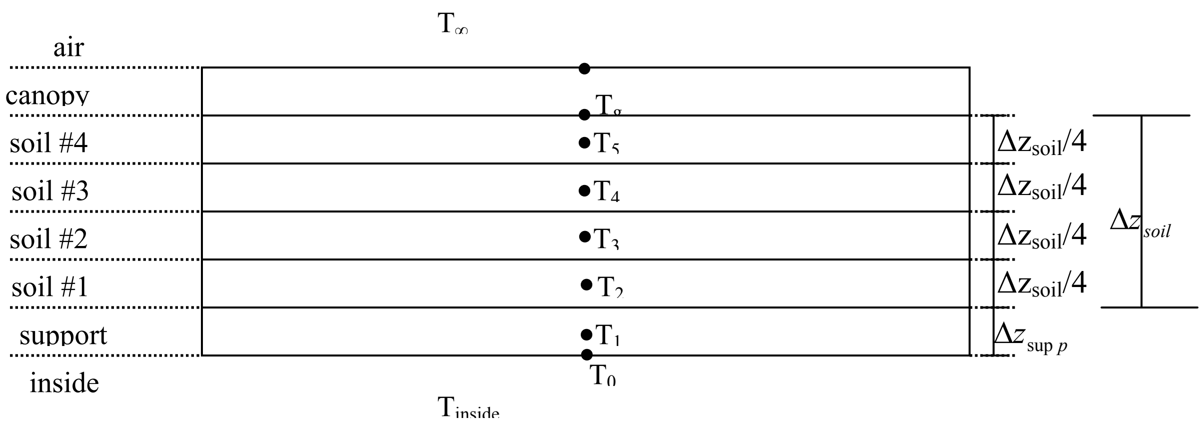


Figure 1: Schematic of modeled green roof system.

This arrangement provides a 1-D heat transfer model, through the vertical axis.

An energy balance for each layer was set up by solving for the energy fluxes, FLUX IN – FLUX OUT = STORAGE. For the first layer, the boundary layer between the inside air and the structural support, there is no storage, so the balance is just the difference between the conductive and convective flux, shown below in Equation 1.

$$\frac{k_{\sup p}}{\Delta z_{\sup p}/2}\left(T_1 - T_0\right) - h_{inside}\left(T_0 - T_{inside}\right) = 0 \qquad (1)$$

An explanation of variables and symbols can be found in Appendix A.

The next five layers (structural support and soil layers 1-4) each include a storage term in the form of, $\rho_i cp_i \Delta z_i(T_j - T_{j,t-1})$, where i is the layer type (support or soil) and j is the layer number. The remaining components for the balance are found by the difference of the conductive flux in and out for each layer. Equations 2-6 below are the energy balances for the structural support layer and the four soil layers.

$$\frac{8k_{soil}k_{\sup p}}{k_{\sup p}\Delta z_{soil} + 4k_{soil}\Delta z_{\sup p}}\left(T_2 - T_1\right) - \frac{2k_{\sup p}}{\Delta z_{\sup p}}\left(T_1 - T_0\right) = \rho_{\sup p}cp_{\sup p}\Delta z_{\sup p}\left(T_1 - T_{1,t-1}\right) \qquad (2)$$

$$\frac{4k_{soil}}{\Delta z_{soil}}\left(T_3 - T_2\right) - \frac{8k_{soil}k_{\sup p}}{k_{\sup p}\Delta z_{soil} + 4k_{soil}\Delta z_{\sup p}}\left(T_2 - T_1\right) = \rho_{soil} cp_{soil} \frac{\Delta z_{soil}}{4}\left(T_2 - T_{2,t-1}\right) \tag{3}$$

$$\frac{k_{soil}}{\Delta z_{soil}/4}\left(T_4 - T_3\right) - \frac{k_{soil}}{\Delta z_{soil}/4}\left(T_3 - T_2\right) = \rho_{soil} cp_{soil} \frac{\Delta z_{soil}}{4}\left(T_3 - T_{3,t-1}\right) \tag{4}$$

$$\frac{k_{soil}}{\Delta z_{soil}/4}\left(T_5 - T_4\right) - \frac{k_{soil}}{\Delta z_{soil}/4}\left(T_4 - T_3\right) = \rho_{soil} cp_{soil} \frac{\Delta z_{soil}}{4}\left(T_4 - T_{4,t-1}\right) \tag{5}$$

$$\frac{k_{soil}}{\Delta z_{soil}/8}\left(T_g - T_5\right) - \frac{k_{soil}}{\Delta z_{soil}/4}\left(T_5 - T_4\right) = \rho_{soil} cp_{soil} \frac{\Delta z_{soil}}{4}\left(T_5 - T_{5,t-1}\right) \tag{6}$$

Equations 2 and 3 differ from the remaining equations because the balances include two different materials, the structural support and the soil. These equations were found by using a thermal resistance analog, spanning both of these materials.

The next balance solved is the boundary between the soil and the canopy layer, or the topmost surface of the soil. Since this balance is over a surface and not a layer, there is no storage, so the balance is just the difference between the FLUX IN and the FLUX OUT. There is one incoming flux, solar radiation, and five outgoing flux's (vapor loss, convection, net thermal radiation out and conduction).

The solar radiation term is the amount of solar radiation that passes through the canopy (i.e. the amount not reflected by the canopy) and is absorbed by the soil. Since the Big Leaf model was used to describe the canopy, the fraction of solar radiation transmitted through the canopy was calculated as e^{-kLAI}. The Big Leaf model assumes that the canopy is one large leaf (measured by Leaf Area Index (LAI)), instead of many smaller plants. LAI is an indicator of canopy density, as it is the ratio of upper leaf surface area divided by the occupied land surface area. Equation 7 below is the solar radiation term for this energy balance.

$$\mathrm{RAD}_{\mathrm{solar,\,in,\,g}} = (1-\rho_c)\phi_s e^{-kLAI}\alpha_{soil} \tag{7}$$

The water vapor loss term was included to take care of the latent heat loss in the system, and includes the difference between saturation vapor pressure at the layer and the partial vapor pressure in the air (Campbell 1998). Equation 8 is the vapor loss term.

$$\text{Vapor Loss}_{\mathrm{g}} = \frac{\lambda g_v\left(pws(T_g) - pw\right)}{P} \tag{8}$$

The thermal radiation out includes two terms, the radiation exchange between the ground and the canopy and the ground and the sky. The Stefan-Boltzmann Law applies to these exchanges, which is defined as $\varepsilon\sigma(T_1^4 - T_2^4)$. To simplify these calculations, the Stefan-Boltzmann Law can be linearized to produce a new equation: $4\varepsilon\sigma T_{ave}^3 \Delta T$. Equations 9 and 10 below are the result of the application of the linearized Stefan-Boltzmann Law to the thermal radiation exchange between ground/canopy and ground/sky respectively.

$$\mathrm{RAD}_{\mathrm{thermal,\,out,\,g\to c}} = 4LAI\varepsilon_g\sigma\left(\frac{T_g + T_c}{2}\right)^3\left(T_g - T_c\right) \tag{9}$$

$$\mathrm{RAD}_{\mathrm{thermal,\,out,\,g\to sky}} = 4(1-LAI)\varepsilon_g\sigma\left(\frac{T_g + T_{sky}}{2}\right)^3\left(T_g - T_{sky}\right) \tag{10}$$

LAI is defined as $LAI = \begin{cases} LAI, LAI < 1 \\ 1, LAI \geq 1 \end{cases}$, only when included in those equations that deal with thermal radiation exchange. The LAI term is included in both of these equations because the thermal radiation from the ground is affected by the presence of the canopy.

Equations 7-10 are combined, along with the conductive/convective losses, to create the balance between the soil and the canopy layer. Equation 11 below is the energy balance on the topmost surface of the soil.

$$(1-\rho_c)\phi_s e^{-kLAI}\alpha_{soil} - 4(1-LAI)\varepsilon_g \sigma\left(\frac{T_g + T_{sky}}{2}\right)^3 \left(T_g - T_{sky}\right) - h_{air}\left(T_g - T_\infty\right)$$

$$-4LAI\varepsilon_g \sigma\left(\frac{T_g + T_c}{2}\right)^3 \left(T_g - T_c\right) - \frac{\lambda g_v\left(pws(T_g) - pw\right)}{P} - \frac{k_{soil}}{\Delta z_{soil}/8}\left(T_g - T_5\right) = 0 \qquad (11)$$

The final balance solved is on the canopy, more specifically the boundary between the canopy and the ambient air. It was assumed that there was no storage within the canopy layer or conduction within the layer. There are two incoming flux's (solar radiation and thermal radiation between ground/canopy) and four outgoing flux's (vapor loss, convection, and thermal radiation out). Many of the terms from Equation 11 remained the same or were slightly modified for this balance, shown below in Equation 12.

$$\phi_s(1-\rho_c)\left(1-e^{-kLAI}\right) + 4LAI\varepsilon_g \sigma\left(\frac{T_g + T_c}{2}\right)^3 \left(T_g - T_c\right) - h_{air}\left(T_c - T_\infty\right) \qquad (12)$$

$$-4\varepsilon_c \sigma\left(\frac{T_c + T_{sky}}{2}\right)^3 \left(T_c - T_{sky}\right) - 4\varepsilon_c \sigma\left(\frac{T_c + T_g}{2}\right)^3 \left(T_c - T_g\right) - \frac{\lambda g_v\left(pws(T_c) - pw\right)}{P} = 0$$

(1-e^{-kLAI}) is the amount of solar radiation that does not get through the canopy. The LAI term was not included in the outward thermal radiation exchange between canopy/sky and canopy/ground because the presence of a canopy does not affect these exchanges.

Equations 1-6 and 11-12 were manipulated algebraically to group terms together with similar temperature dependence. Since these equations must be solved simultaneously, they were then formed into a matrix equation of the form below (Equation 13).

$$
\begin{bmatrix}
a_{11} & a_{12} & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{21} & a_{22} & a_{23} & 0 & 0 & 0 & 0 & 0 \\
0 & a_{32} & a_{33} & a_{34} & 0 & 0 & 0 & 0 \\
0 & 0 & a_{43} & a_{44} & a_{45} & 0 & 0 & 0 \\
0 & 0 & 0 & a_{54} & a_{55} & a_{56} & 0 & 0 \\
0 & 0 & 0 & 0 & a_{65} & a_{66} & a_{67} & 0 \\
0 & 0 & 0 & 0 & 0 & a_{76} & a_{77} & a_{78} \\
0 & 0 & 0 & 0 & 0 & 0 & a_{87} & a_{88}
\end{bmatrix}
\times
\begin{bmatrix} T_0 \\ T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_g \\ T_c \end{bmatrix}
=
\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \\ c_8 \end{bmatrix}
\tag{13}
$$

Where:

$$a_{11} = \frac{k_{\sup p}}{\Delta z_{\sup p}/2} + h_{inside},$$

$$a_{12} = \frac{-k_{\sup p}}{\Delta z_{\sup p}/2},$$

$$a_{21} = \frac{-k_{\sup p}}{\Delta z_{\sup p}/2},$$

$$a_{22} = \frac{2k_{\sup p}}{\Delta z_{\sup p}/2} + \rho_{\sup p} cp_{\sup p} \Delta z_{\sup p},$$

$$a_{23} = \frac{-k_{soil}}{\Delta z_{soil}/8},$$

$$a_{32} = \frac{-k_{\sup p}}{\Delta z_{\sup p}/2},$$

$$a_{33} = \frac{3k_{soil}}{\Delta z_{soil}/4} + \rho_{soil} cp_{soil} \frac{\Delta z_{soil}}{4},$$

$$a_{34} = \frac{-k_{soil}}{\Delta z_{soil}/4},$$

$$a_{43} = \frac{-k_{soil}}{\Delta z_{soil}/4},$$

$$a_{44} = \frac{2k_{soil}}{\Delta z_{soil}/4} + \rho_{soil} cp_{soil} \frac{\Delta z_{soil}}{4},$$

$$a_{45} = \frac{-k_{soil}}{\Delta z_{soil}/4},$$

$$a_{54} = \frac{-k_{soil}}{\Delta z_{soil}/4},$$

$$a_{55} = \frac{2k_{soil}}{\Delta z_{soil}/4} + \rho_{soil} cp_{soil} \frac{\Delta z_{soil}}{4},$$

$$a_{56} = \frac{-k_{soil}}{\Delta z_{soil}/4},$$

$$a_{65} = \frac{-k_{soil}}{\Delta z_{soil}/4},$$

$$a_{66} = \frac{3k_{soil}}{\Delta z_{soil}/4} + \rho_{soil} cp_{soil} \frac{\Delta z_{soil}}{4},$$

$$a_{67} = \frac{-k_{soil}}{\Delta z_{soil}/8},$$

$$a_{76} = \frac{-k_{soil}}{\Delta z_{soil}/8},$$

$$a_{77} = 4\varepsilon_g \sigma \left((1 - LAI)\left(\frac{T_g + T_{sky}}{2}\right)^3 + LAI\left(\frac{T_g + T_c}{2}\right)^3 \right) + h_{air} + \frac{k_{soil}}{\Delta z_{soil}/8},$$

$$a_{78} = -4LAI\varepsilon_g \sigma \left(\frac{T_g + T_c}{2}\right)^3,$$

$$a_{87} = -4\sigma \left(\frac{T_g + T_c}{2}\right)^3 \left(LAI\varepsilon_g + \varepsilon_c\right),$$

$$a_{88} = 4\sigma \left(LAI\varepsilon_g \left(\frac{T_g + T_c}{2}\right)^3 + \varepsilon_c \left(\left(\frac{T_c + T_{sky}}{2}\right)^3 + \left(\frac{T_c + T_g}{2}\right)^3 \right) \right) + h_{air},$$

$$c_1 = h_{inside} T_{inside},$$

$$c_2 = \rho_{\sup p} cp_{\sup p} \Delta z_{\sup p} T_{1,t-1},$$

$$c_3 = \rho_{soil} cp_{soil} \frac{\Delta z_{soil}}{4} T_{2,t-1},$$

$$c_4 = \rho_{soil} cp_{soil} \frac{\Delta z_{soil}}{4} T_{3,t-1},$$

$$c_5 = \rho_{soil} cp_{soil} \frac{\Delta z_{soil}}{4} T_{4,t-1},$$

$$c_6 = \rho_{soil} cp_{soil} \frac{\Delta z_{soil}}{4} T_{5,t-1},$$

$$c_7 = (1 - \rho_c)\phi_s e^{-kLAI} \alpha_{soil} + h_{air} T_\infty + 4(1 - LAI)\varepsilon_g \sigma \left(\frac{T_g + T_{sky}}{2}\right)^3 T_{sky} - \frac{\lambda g_v pws(T_g)}{P} + \frac{\lambda g_v pw}{P},$$

$$c_8 = \phi_s(1-\rho_c)(1-e^{-kLAI}) + h_{air}T_\infty + 4\varepsilon_c\sigma\left(\frac{T_c + T_{sky}}{2}\right)^3 T_{sky} - \frac{\lambda g_v pws(T_c)}{P} + \frac{\lambda g_v pw}{P}.$$

The temperature profile of the system can be solved now by inverting the 8x8 matrix and multiplying by the solution matrix (c_1, c_2,…, c_8).

Parameters Used

The creation of the energy balance model led for the need to create additional parameters so a solution could be found. These parameters were vapor conductance (g_v), sky temperature (T_{sky}), and the convection coefficient of air (h_{air}). Vapor conductance was found to be dependent on wind speed and is shown below in Equation 14 (Campbell 1998).

$$g_v = \left(\frac{0.6 \times 0.2u}{0.6 + 0.2u}\right) 3600 \tfrac{s}{hr} \tag{14}$$

The 3600 s/hr was included in the equation to convert from seconds to hour. Since the model runs on an hourly basis, all parameters and terms must be hourly. This convention was maintained throughout the simulation process.

The sky temperature parameter, normally only a function of air temperature was modified to allow for cloudy sky conditions, which can affect the sky temperature. Equation 15 below provides the updated sky temperature equation that is dependent on air temperature and cloud cover.

$$T_{sky} = T_\infty - \left(T_\infty - 0.0552T_\infty^{1.5}\right)(1 - CC) \tag{15}$$

The fraction of cloud cover was included into the original sky temperature equation, so that when there was full cloud cover, $T_{sky} = T_{\infty}$, and when there was no cloud cover $T_{sky} = 0.0552T_{\infty}^{1.5}$, the original Swinbank equation for sky temperature (Albright 1990).

The convection coefficient of air was derived empirically from seasonal data. Winter and summer values of h_{air} were found along with their relative wind speeds, and were converted to their appropriate units (wind speed in m/s and h_{air} in W/m^2K) (Leckie, Masters, et al. 1981). Using the general formula of $h_{air} = A + uB$ (where u is wind speed), two equations were formed for summer and winter conditions, and both A and B were solved for. Table 1 below provides values for h_{air} and u for both seasons.

Table 1: Seasonal values for convection coefficient of air and wind speed (Leckie, Masters, et al. 1981).

	h_{air} (W/m^2K)	u (m/s)
Winter	34	6.7
Summer	23	3.4

The resulting equation, shown below in Equation 16, allows for the convection coefficient of air to be solved for any give wind speed, regardless of season.

$$h_{air} = (11.67 + 3.33u)3600\tfrac{s}{hr} \tag{16}$$

Units for A are in W/m^2K and B are in J/m^3K.

Weather Data

Weather data was taken from The American Society of Heating, Refrigerating and Air Conditioning Engineers' (ASHRAE) Technical Committee 4.2: Weather Information CD-ROM. Hourly weather data was extracted from the Weather Year for Energy Calculations (WYEC) files. ASHRAE has collected and compiled 77 data files for locations across the United States and Canada; however only 3 locations were chosen for simulation: New York, NY, Phoenix, AZ, and Santa Maria, CA. These locations were chosen for their different climates: cold, wet (NY-temperate); hot, dry (AZ-desert); mild, semi-arid (CA – Mediterranean).

Weather files for each of these locations were imported into a spreadsheet and all extraneous data fields were removed. The six data fields kept were: date/hour, global horizontal irradiance (kJ/m^2), dry bulb temperature (°C), dew point (°C), wind speed (m/s), and opaque sky cover. Gaps in the data sets were filled by observing trends preceding and succeeding the gaps, and matching values as close as possible. Most of the gaps were present in the opaque sky cover field.

In addition to the imported data, saturation vapor pressure and partial vapor pressure were calculated to solve for relative humidity. Hourly dry bulb temperatures were converted to Kelvin to determine saturation vapor pressure in Pascals, using Equation 17 below (Albright 1990):

$$pws = \exp\left(A_1/T + A_2 + A_3T + A_4T^2 + A_5T^3 + A_6T^4 + A_7\ln(T)\right) \tag{17}$$

The coefficients A_1-A_7 have the following values shown in Table 2 below.

Table 2: Coefficients A_1-A_7 used in Equation 14.

	Over Ice	**0°C ≤ T ≤ 200°C**
A_1	-5.6745359×10^{3}	-5.8002206×10^{3}
A_2	6.3925247	1.3914993
A_3	-9.677843×10^{-3}	$-48.640239 \times 10^{-3}$
A_4	0.622157×10^{-6}	41.764768×10^{-6}
A_5	2.0747825×10^{-9}	$-14.452093 \times 10^{-9}$
A_6	$-0.9484024 \times 10^{-12}$	0.0
A_7	4.1635019	6.5459673

Partial vapor pressure can be solved by using the fitted equation found in the ASHRAE Handbook of Fundamentals, which relates the dew point temperature to the partial vapor pressure, in Equation 18.

$$\begin{aligned} &\text{For } -60°C \le t_d < 0°C: \quad t_d = -60.45 + 7.0322\ln(pw) + 0.37(\ln(pw))^2 \\ &\text{For } 0°C \le t_d < 70°C: \quad t_d = -35.957 - 1.8726\ln(pw) + 1.1689(\ln(pw))^2 \end{aligned} \tag{18}$$

Solving for the positive root of Equation 18 provides the partial vapor pressure. The hourly relative humidity can now be solved for by dividing the hourly partial vapor pressure by the saturation vapor pressure, shown below in Equation 19.

$$rH = \frac{pw}{pws} \tag{19}$$

Following these calculations, data fields were then converted to their appropriate units, temperatures to Kelvin and global horizontal irradiance to J/m^2. Additionally each data set was condensed to include only the growing season for each location. The full years data was used for the AZ and CA simulations, while only Julian Dates 91-304 (April 1 – October 31) were used for NY. The fields that were needed to run the simulation were placed in another spreadsheet to be exported as text files. The six fields that were exported were: hour, global horizontal irradiance (J/m^2), dry bulb temperature (K), wind speed (m/s), opaque sky cover, and relative humidity. Appendix C contains the first hour's data for the three text files that contain the weather data used in the simulations.

Program

In order to solve the matrix equation shown in Equation 13 a computer program had to be developed, that accomplished the following tasks:

1. Asks user to choose location
2. Receives weather data from text files based on choice
3. Creates arrays from the fields of text files
4. Creates output text file
5. Loops for each day
    a. Loop for each hour
        i. Extracts hourly data from data arrays
        ii. Calculates hourly parameters (g_v, T_{sky}, h_{air})
        iii. Loop 3 times for Stefan Boltzmann Law linearization convergence
            1. Calculates variables in matrix equation
            2. Solves for temperatures in system
            3. Recalculates cubed average temperature terms
        iv. Ends Stefan-Boltzmann Law linearization loop
    b. Extracts hourly temperature from temperature array to output file
    c. Ends hour loop
6. Ends day loop
7. Ends program

The above algorithm was placed into code in the MATrix LABoratory (MATLAB) computer language environment. The program created, named GreenRoofModel.m, and all supporting functions, can be found in Appendix D. Figure 2 below shows a visual representation of how the functions interact with the main program.

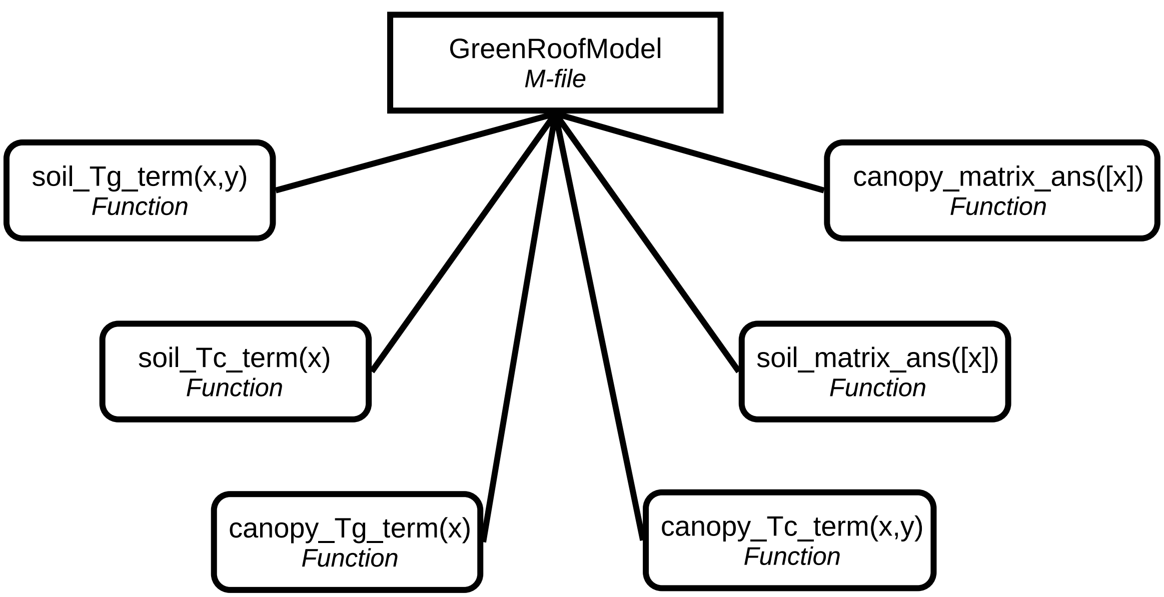


Figure 2: Representation of interaction between main program and supporting functions.

The functions were created to deal with those equations that had cubed average temperature terms. The notation *function_name*(x) and *function_name*(x,y) indicates that the function receives one or two cubed average temperatures from the main program; whereas *function_name*([x]) indicates that it receives an array of data from the main program. Each function can be matched to a variable within the matrix equation (Equation 13), shown below in Table 3.

Table 3: Variables from Equation 13 and their representative function names.

soil_Tg_term(x,y):	a_{77}
soil_Tc_term(x):	a_{78}
canopy_Tg_term(x):	a_{87}
canopy_Tc_term(x,y):	a_{88}
soil_matrix_ans([x]):	c_7
canopy_matrix_ans([x]):	c_8

The created program first defines all variables and parameters used in the program. These values can be found in Appendix B. There are six parameters that were set to vary for a sensitivity analysis: thermal diffusivity of the soil, thermal conductivity of the soil, depth of soil layer, LAI, reflectivity of the soil, and thermal conductivity of the support structure. Since there was a lack of data for the density and specific heat of the modeled soil, thermal diffusivity of the soil was used instead. As indicated by Equation 20 below, thermal conductivity divided by thermal diffusivity results in the volumetric heat capacity, or the product of density and specific heat, needed in Equations 3-6.

$$\rho cp = \frac{k}{\alpha} \tag{20}$$

Initial conditions were then defined to start the execution of the program. Since the program calculates hourly values over a large time span, the values of the initial conditions do not affect the outcome of the long term running of the program.

The user is then prompted by a menu to choose the location of the green roof, New York City, NY, Phoenix, AZ, or Santa Maria, CA. Based on the user's choice, the six fields from the weather data files are placed into separate arrays. Next, the output text file is created and opened so the solved hourly temperature values can be exported.

To begin the hourly calculations, three **for** loops are utilized: one for each day being modeled, one for each hour in a day, and one for the convergence of the Stefan-Boltzmann Law linearization. The linearization loop is needed because the cubed average temperature calculation is first solved using the previous hour's temperatures, and not the current hour's temperatures that are being solved for. It was assumed that if these calculations were looped three times that the values would converge, eliminating the discrepancy between the previous hour's temperature and the current temperature.

Within the linearization loop, all the values needed for the matrix equation (Equation 13) are solved, along with all supporting functions. The functions just calculate their appropriate equations indicated in Table 3; however soil_matrix_ans and canopy_matrix_ans also calculate the saturated and partial vapor pressure at the previous hour's soil and canopy temperature. The functions return the variables to the main program and the hourly temperature array is solved for by inverting the 8x8 variable matrix and multiplying it by the solution matrix (c_1, c_2,…, c_8). The linearization loop is then ended, and the current hour's temperatures are stored and exported to the output file.

The remaining loops are closed and the program's execution is completed after every hour is solved for.

Results and Discussion

Model Validation

It is necessary whenever a new model is created to prove that it produces reasonable results before running the actual simulations. This model validation consisted of supplying the program with the same 24 hour weather data set for ten consecutive days. The first day's worth of data from the New York, NY weather file was used as the test set, shown below in Table 4.

Table 4: Weather data used in model validation. First 24 hour's of NY weather file applied consecutively for ten days.

Hour	Global Horizontal Irradiance (J/m^2)	Dry Bulb Temperature (K)	Wind Speed (m/s)	Opaque Sky Cover	Relative Humidity
1	0	279.65	6.5	1.0	0.66
2	0	279.35	7.2	1.0	0.66
3	0	279.15	7.6	1.0	0.66
4	0	279.05	7.9	1.0	0.66
5	0	279.15	7.9	1.0	0.66
6	0	279.35	7.6	1.0	0.65
7	190000	279.25	6.7	1.0	0.65
8	485000	280.15	5.8	1.0	0.61
9	1485000	281.15	5.0	1.0	0.58
10	2013000	282.05	4.1	0.8	0.56
11	2101000	282.95	5.1	0.8	0.54
12	2337000	283.95	6.2	0.6	0.52
13	2724000	284.85	7.2	0.3	0.50
14	2242000	284.45	7.0	0.6	0.53
15	863000	284.15	6.9	0.8	0.57
16	578000	283.75	6.7	1.0	0.61
17	441000	282.95	5.8	1.0	0.65
18	117000	282.25	5.0	0.8	0.69
19	4000	281.45	4.1	0.8	0.74
20	0	281.85	3.4	0.7	0.70
21	0	282.15	2.8	0.7	0.67
22	0	282.55	2.1	0.6	0.63
23	0	281.85	2.1	0.6	0.67
24	0	281.25	2.1	0.7	0.71

The purpose of running the same data set for ten days is to see if the solved temperatures approach a diurnal cycle, or the same daily temperature variation throughout each day. Figures 3-6 below show the daily temperature variation in the roof using the above data.

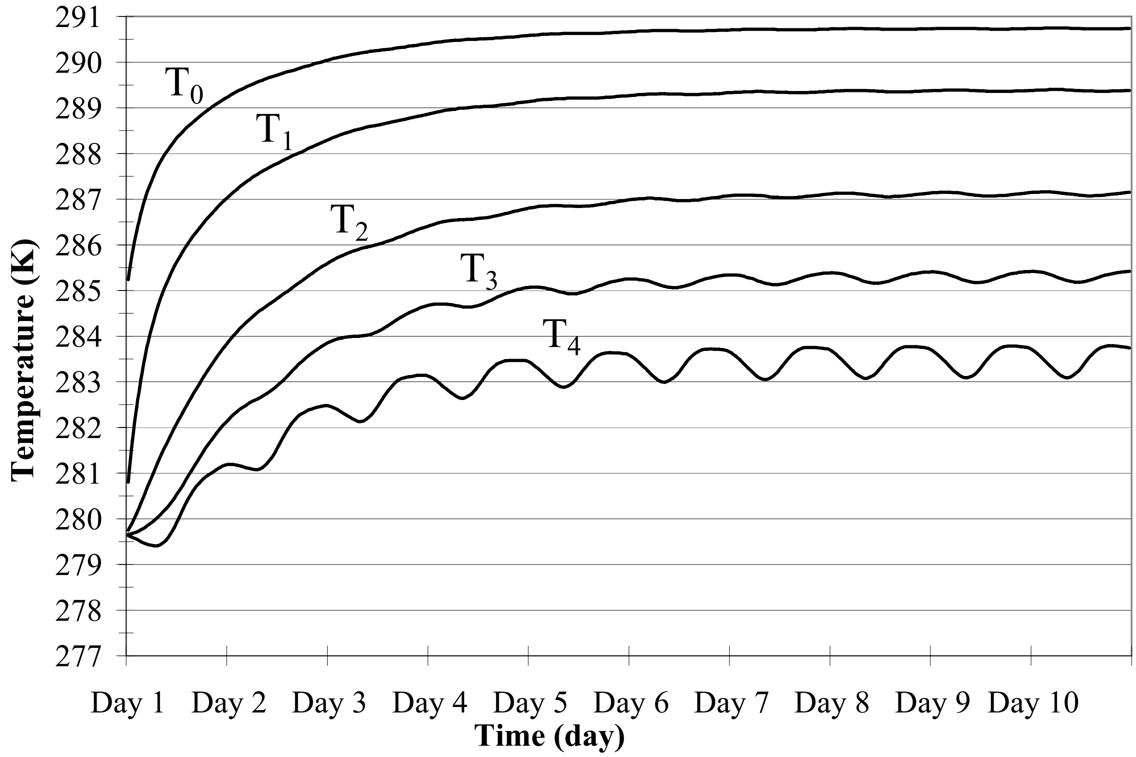


Figure 3: Temperature variation throughout model validation for T_0-T_4.

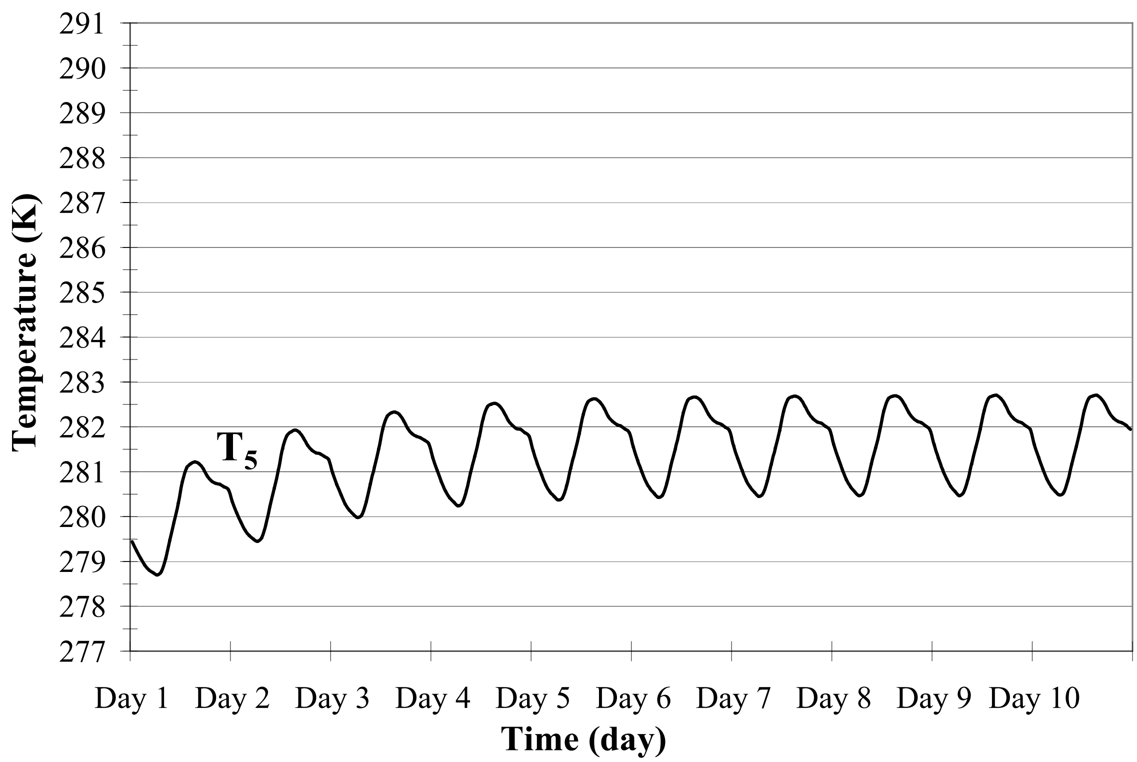


Figure 4: Temperature variation throughout model validation for T_5.

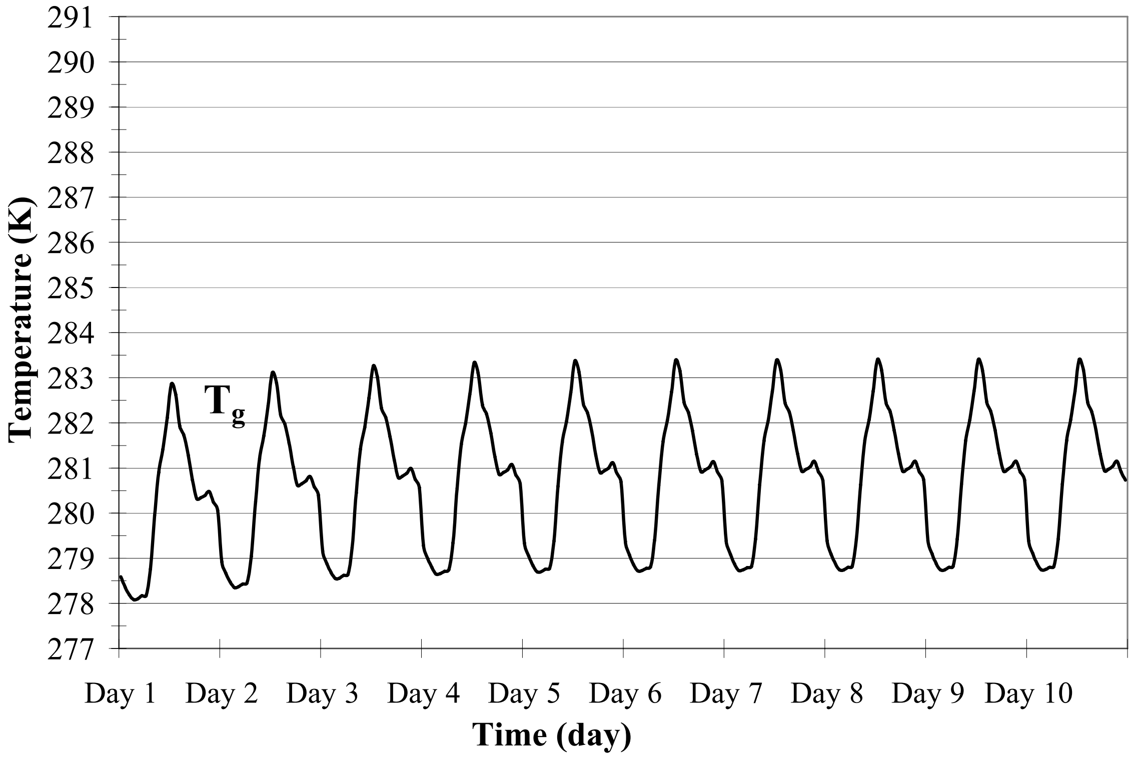


Figure 5: Temperature variation throughout model validation for T_g.

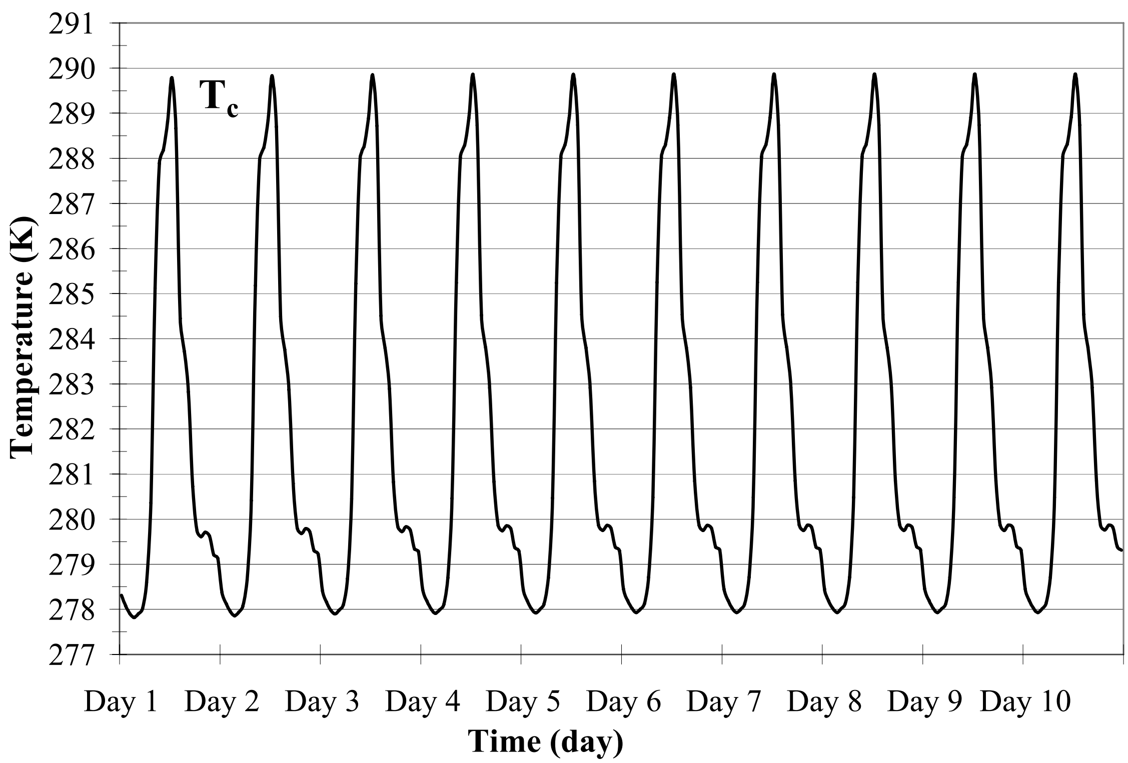


Figure 6: Temperature variation throughout model validation for T_c.

A nice diurnal temperature variation is exhibited in the layers T_3-T_c, shown in Figures 3-6; however the first three layers (T_0-T_2) don't appear to exhibit such a cycle and seem to converge to one value. T_{inside} is kept at a constant value, which causes the T_0-T_2 graphs to exhibit this non-diurnal cycle. The upper layers are affected more by the varying air temperature and solar radiation, which causes the diurnal cycles exhibited above in Figure 3-6. Table 5 below shows the amount of time needed for equilibration and the approximate temperatures at which each layer approached or oscillated around (if a diurnal cycle was exhibited). On average it took about 5 days for the green roof system to approach a diurnal cycle or equilibrium, which is approximately how long a thermal mass of soil takes to equilibrate.

Table 5: Layer analysis for model validation. Temperature (K) and Time (days) for equilibrium/diurnal cycle extrapolated from Figures 3-6.

Layer	Equilibrium/Diurnal	Time (days)	Temperature (K)
T_0	Equilibrium	6	291
T_1	Equilibrium	6	289
T_2	Equilibrium	5	287
T_3	Diurnal	5.5	285
T_4	Diurnal	5	283
T_5	Diurnal	5.5	282
T_g	Diurnal	4	281
T_c	Diurnal	1	284

Sensitivity Analysis

A sensitivity analysis was performed to see how certain parameters affect the performance of a green roof system. There were six parameters that were investigated for the sensitivity analysis: thermal conductivity of the structural support, reflectivity of the soil, LAI, depth of the soil layer, thermal conductivity of the soil, and thermal diffusivity of the soil. Appendix B provides these parameter values, as well as if the parameters were BASE, MAX, MIN, or MID values. Base values were considered to be the standard data set and were used to solve the remaining simulations (Temperature Profile and Flux Analysis). For each parameter analysis, base values were used for the five parameters not being investigated. The sensitivity analysis was conducted on the three locations modeled and these results can be found in Appendix E.

The first set of parameters investigated was the thermal conductivity of the structural support, or whether the roof would or would not be insulated. It was assumed that the roof would be a non-insulated concrete support structure (BASE), and that the insulated simulation would be 10% of the base value, rather than a separate insulation layer. For the NY and AZ simulations, differences were noticed in the T_0-T_3 layers (T_0-T_4 in AZ) during the beginning and ending months of the simulations. These differences are consistent with the fact that the presence of insulation has a greater affect during these colder periods. For the NY simulation, as Figure 7 shows below, the insulation temperature is greater than the non-insulated temperature for layer T_0.

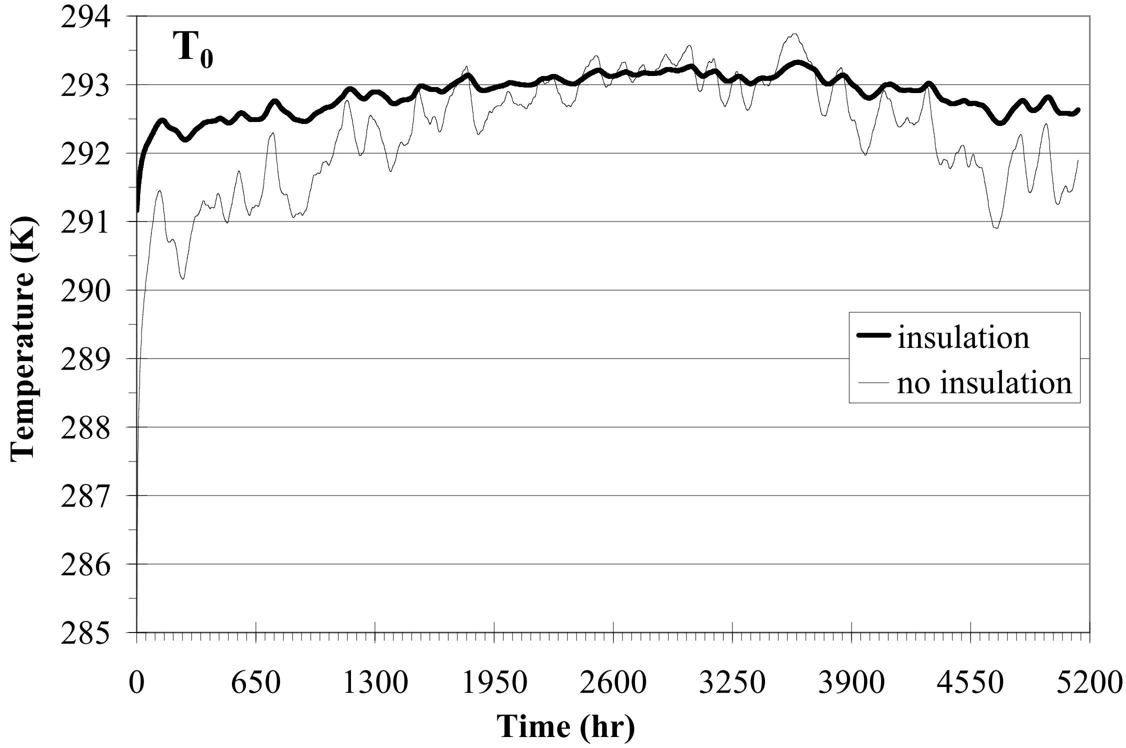


Figure 7: Thermal conductivity of the structural support sensitivity analysis for the NY simulation, layer T_0.

With the presence of insulation the temperature should, and does, remain close to the constant indoor air temperature of 293K. This relationship changes in layers T_1-T_3 (T_1-T_4 for AZ), as the non-insulated simulations have higher temperatures than the insulated ones, shown below in Figure 8.

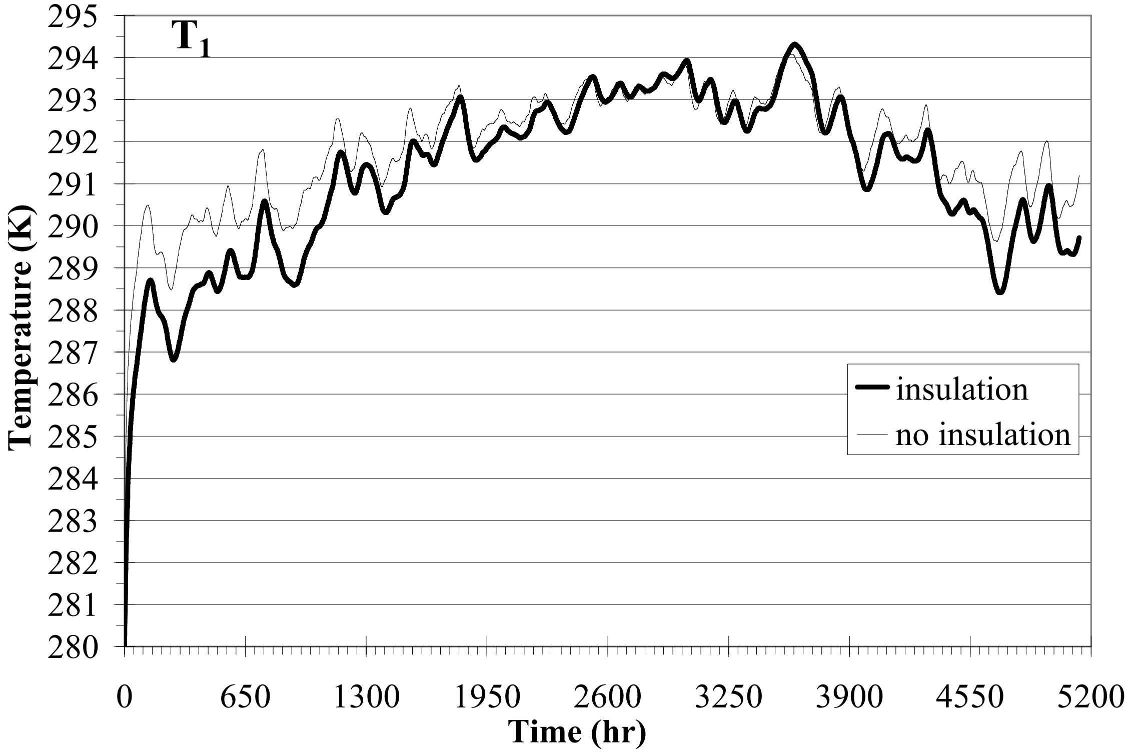


Figure 8: Thermal conductivity of the structural support sensitivity analysis for the NY simulation, layer T_1.

This change can be attributed to the fact that the affect of insulation is only found in the value of k_{supp}, which mostly affects the lowest layers of the simulation (T_0). The CA simulations produced completely different results for this sensitivity analysis, as the insulated/non-insulated plots never cross paths, which reflects the Mediterranean climate of Santa Maria, CA. Figures 9 and 10 below show the sensitivity analysis plots for layers T_0 and T_1 respectively. Note how the insulated simulation always is greater than the non-insulated simulation for T_0, and how that changes for the T_1 layer (T_1-T_4 non-insulated temperatures > insulated temperatures; T_5-T_c no noticeable differences).

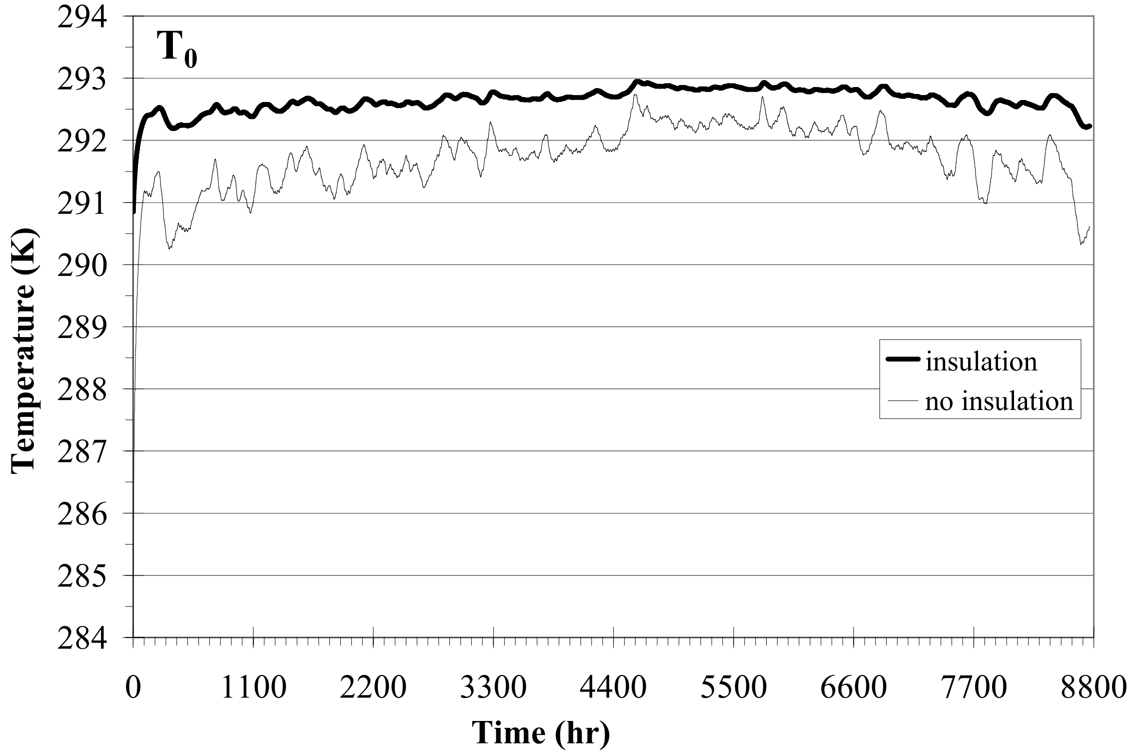


Figure 9: Thermal conductivity of the structural support sensitivity analysis for the CA simulation, layer T_0.

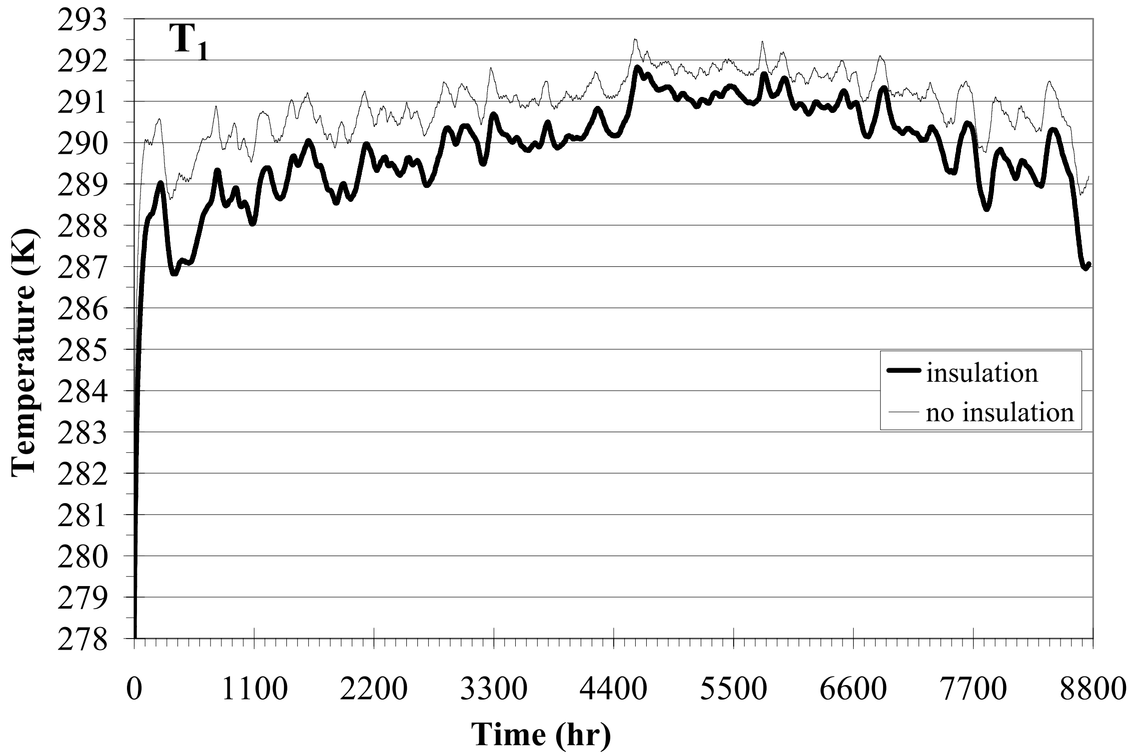


Figure 10: Thermal conductivity of the structural support sensitivity analysis for the CA simulation, layer T_1.

The second set of parameters investigated was the reflectivity of the soil. The soil was assumed to be wet, dark soil (BASE) which was compared to dry, light soil (MID) and dry, white sand (MAX). There weren't any significant differences between the different simulations, and changing the soil reflectivity did not produce a large effect, as the greatest difference between the simulations was 1 K.

LAI was next to be investigated, and similar trends were found for all three locations modeled. In layers T_0-T_g the minimum value of LAI has the highest temperatures throughout the simulation period, followed next by the base value of LAI, and finally the maximum value of LAI ($T_{LAI,\ min} > T_{LAI,\ base} > T_{LAI,\ max}$). This relationship is shown below in Figure 11.

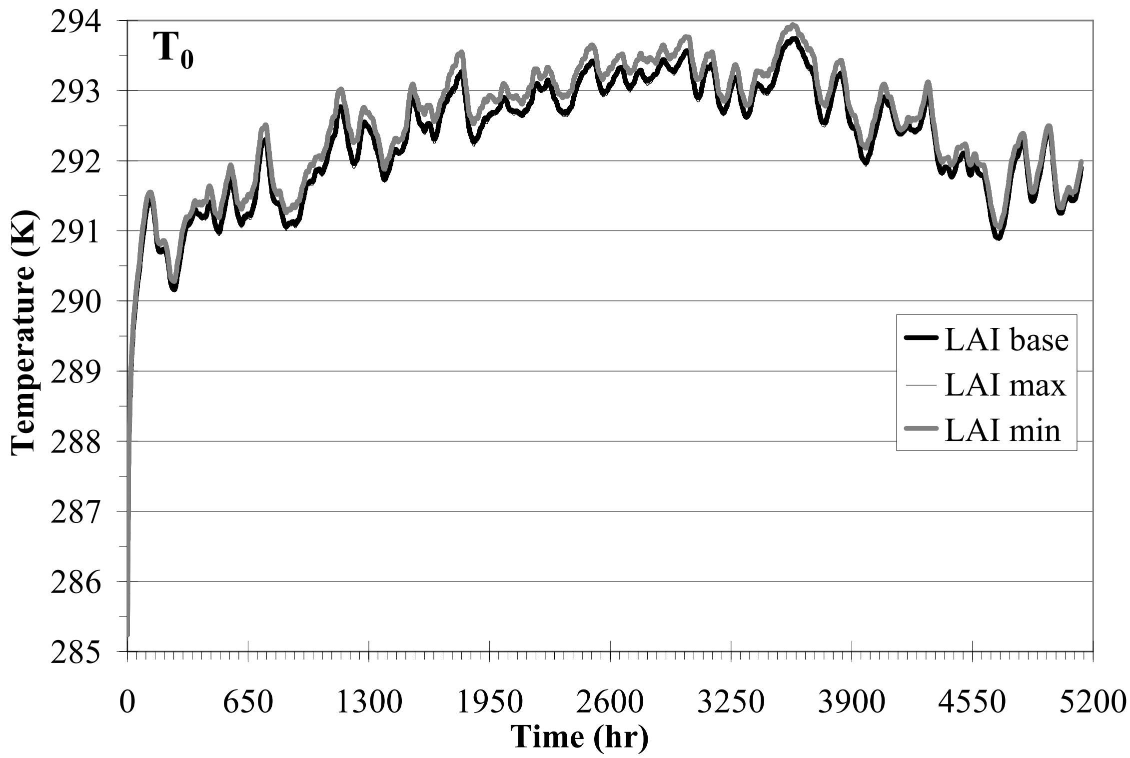


Figure 11: LAI sensitivity analysis for the NY simulation, layer T_0.

A smaller value of LAI indicates less area covered by leaves, and a higher temperature should result, since more solar radiation is hitting the surface of the soil and there is less shading being provided by the plants. The T_c layer shows an opposite relationship as the maximum value of LAI now exhibits the highest temperatures throughout the simulations. A larger LAI value should, and does, affect the canopy temperature since larger canopies have more area for radiation and moisture exchanges. Figure 12 below shows this trend in the T_c layer.

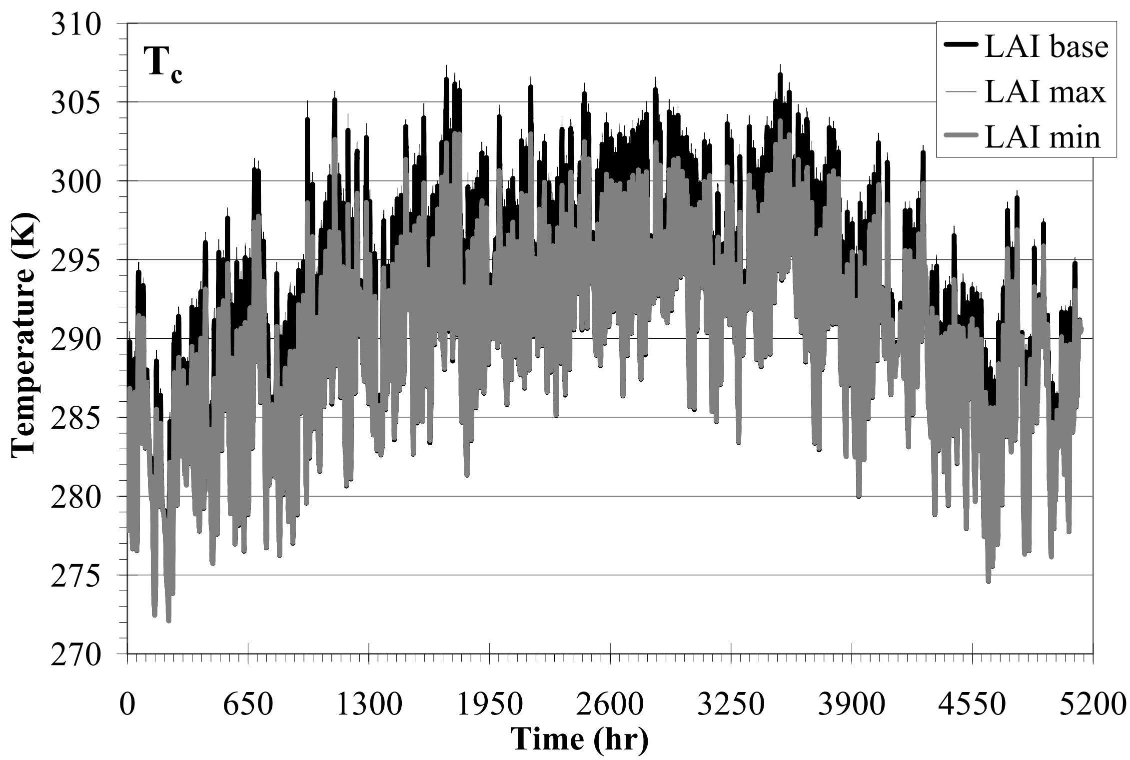


Figure 12: LAI sensitivity analysis for the NY simulation, layer T_c.

The fourth parameter to be investigated was the depth of the soil layer, which was allowed to vary from 0.25 m to 1.0 m, with 0.5 m being the base value. All simulations followed the same trends, but had different values and/or different amounts of variability or noise. For the NY simulation, the maximum value of soil depth for the T_0-T_3 layers all exhibited less variability and greater thermal stability, which is consistent with the larger thermal mass which is produced with more soil. The plot of the maximum value of soil depth appeared to follow the average values of the minimum and the base values in layers T_0-T_4 during the summer months, shown below in Figure 13. For the AZ simulation, this trend appeared only in layers T_0-T_2 due to the desert climate of AZ.

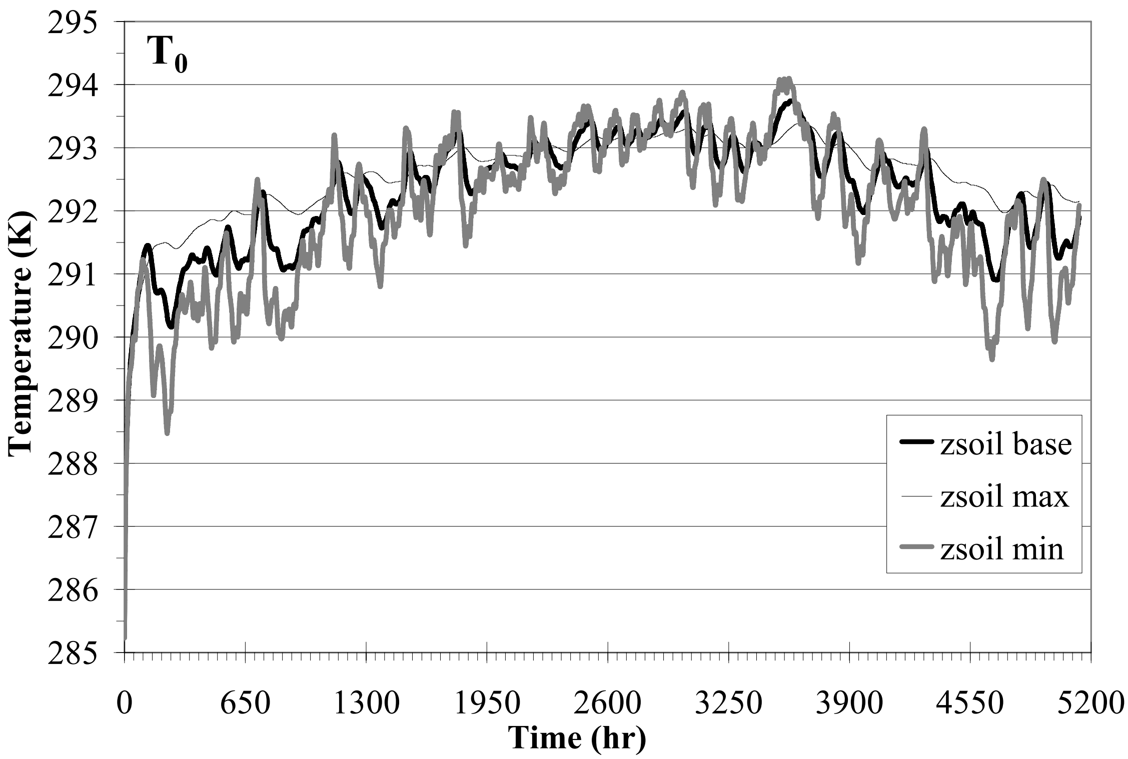


Figure 13: Soil depth sensitivity analysis for the NY simulation, layer T_0.

The CA simulations showed a different trend in the first 3 layers (T_0-T_2) as the maximum soil depth had the highest temperature, followed by the base soil depth and then the minimum soil depth ($T_{soil,\ max} > T_{soil,\ base} > T_{soil,\ min}$), shown below in Figure 14.

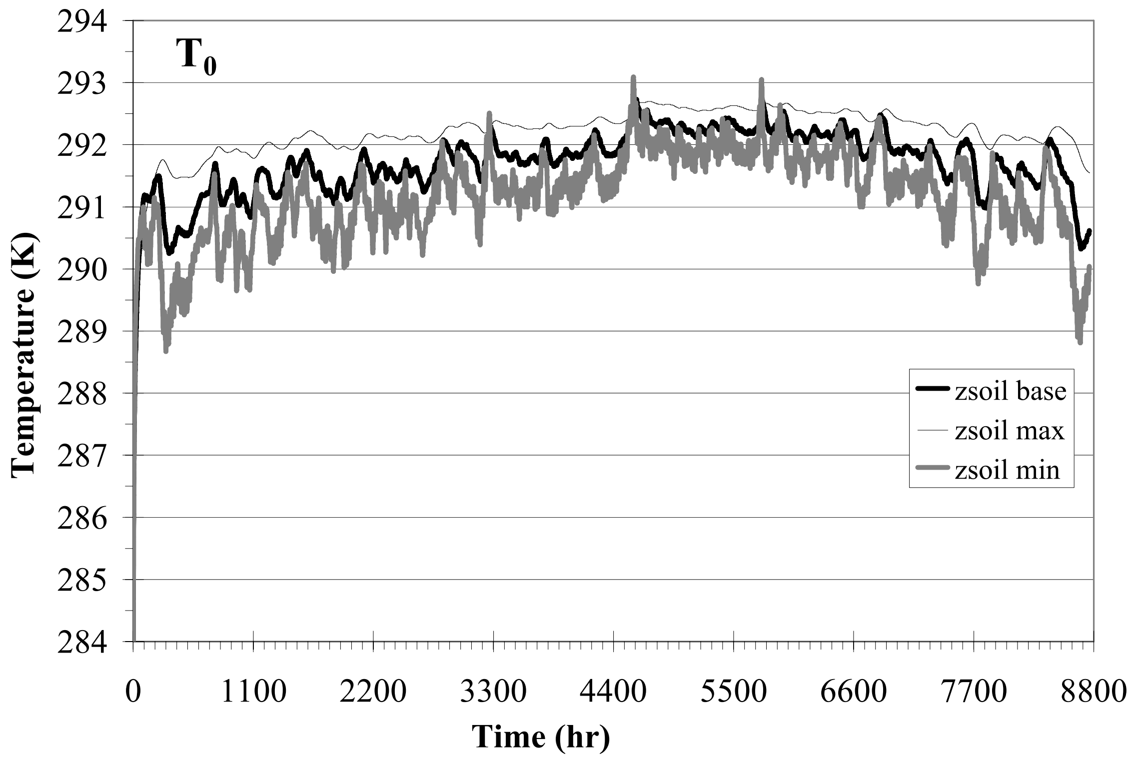


Figure 14: Soil depth sensitivity analysis for the CA simulation, layer T_0.

In the NY simulation, all the curves in the T_5-T_c layers were identical, with very few significant differences (T_3-T_c in both CA and AZ). Since soil depth does not greatly affect these layers (no thermal storage in T_g and T_c), the above trends are justified.

The thermal conductivity of the soil was the next parameter to be investigated. For the NY simulations there were differences in the beginning and ending months for the T_0-T_4 layers, and no significant differences in the summer months. The plot for the minimum value for thermal conductivity of the soil had the highest temperature, followed by the base value and then the maximum value ($T_{k,\ min} > T_{k,\ base} > T_{k,\ max}$), which makes sense since a larger thermal conductivity would result in more conduction occurring, resulting in lower temperatures. Figure 15 below shows the sensitivity analysis for the first layer (T_0) of the NY simulation for thermal conductivity of the soil.

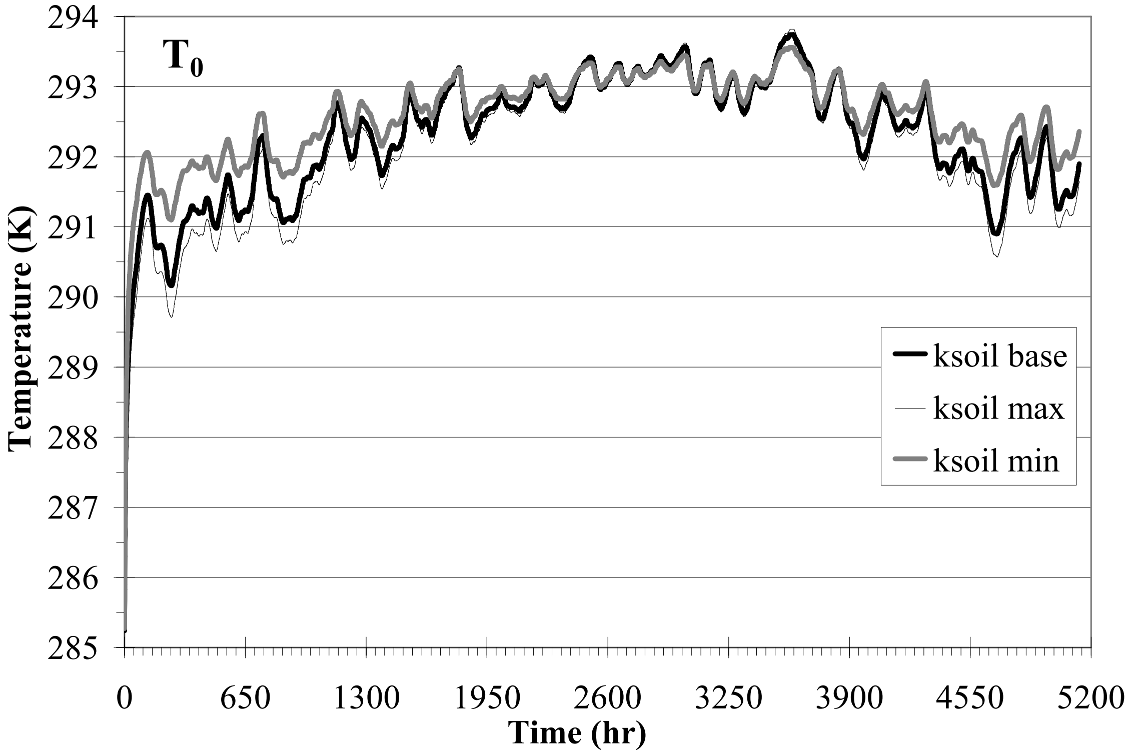


Figure 15: Thermal conductivity of the soil sensitivity analysis for the NY simulation, layer T_0.

For the CA simulations, once again $T_{k,\ min} > T_{k,\ base} > T_{k,\ max}$, however this trend continued for the entire simulation period, or the full year, in layers T_0-T_3. This difference, shown below in Figure 16, is once again due to the Mediterranean climate that is present in Santa Maria, CA.

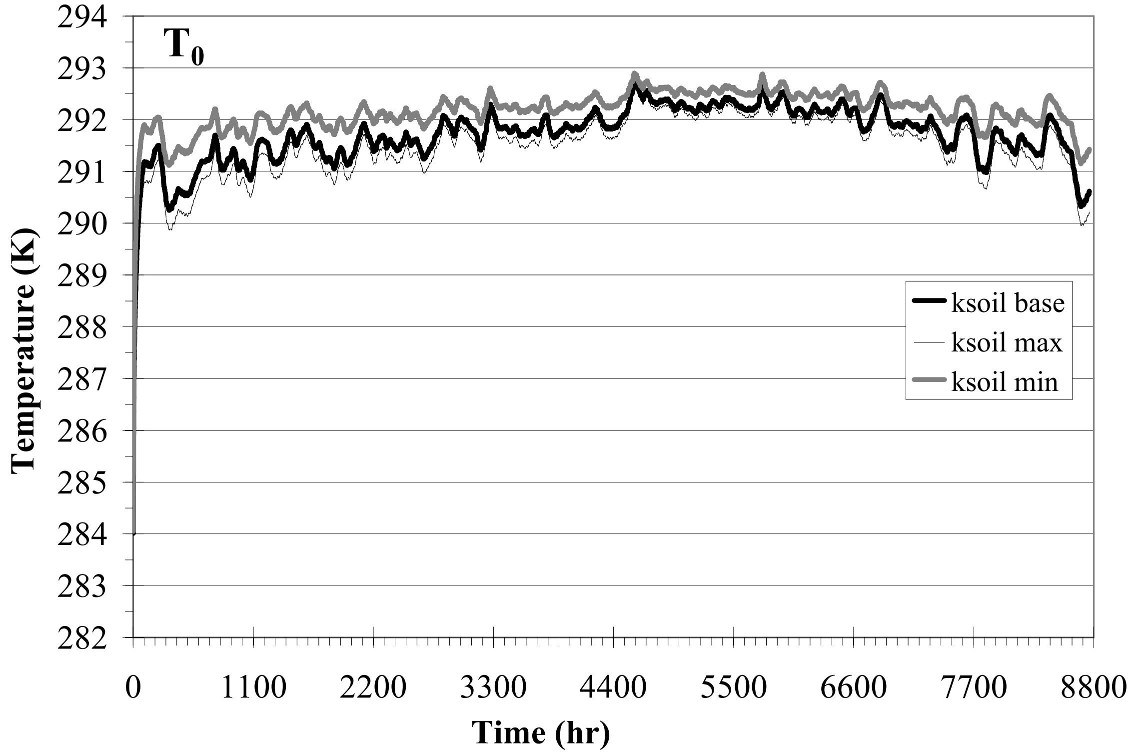


Figure 16: Thermal conductivity of the soil sensitivity analysis for the CA simulation, layer T_0.

The final set of parameters investigated was the thermal diffusivity of the soil. There weren't any significant differences between the different simulations, and changing the thermal diffusivity value of the soil did not produce a large effect. The graphs of this simulation, and the other simulations for the sensitivity analysis, can all be found in Appendix E.

Temperature Profile

Temperature profiles for the three geographic locations were calculated to investigate how the temperature varies moving from the inside of the structure to the outside ambient air. A random date, July 17 (Julian Day 198), was chosen since it is approximately the middle of the NY data set. Using this day's data, temperature profiles were generated for the three locations simulated in six hour intervals: 12:00 AM, 6:00 AM, 12:00 PM and 6:00PM.

For the NY temperature profile, shown below in Figure 17, each investigated time step has approximately the same temperature until layer T_2. For all times, the temperature in the first three layers remains around the constant value of T_{inside}, 293 K.

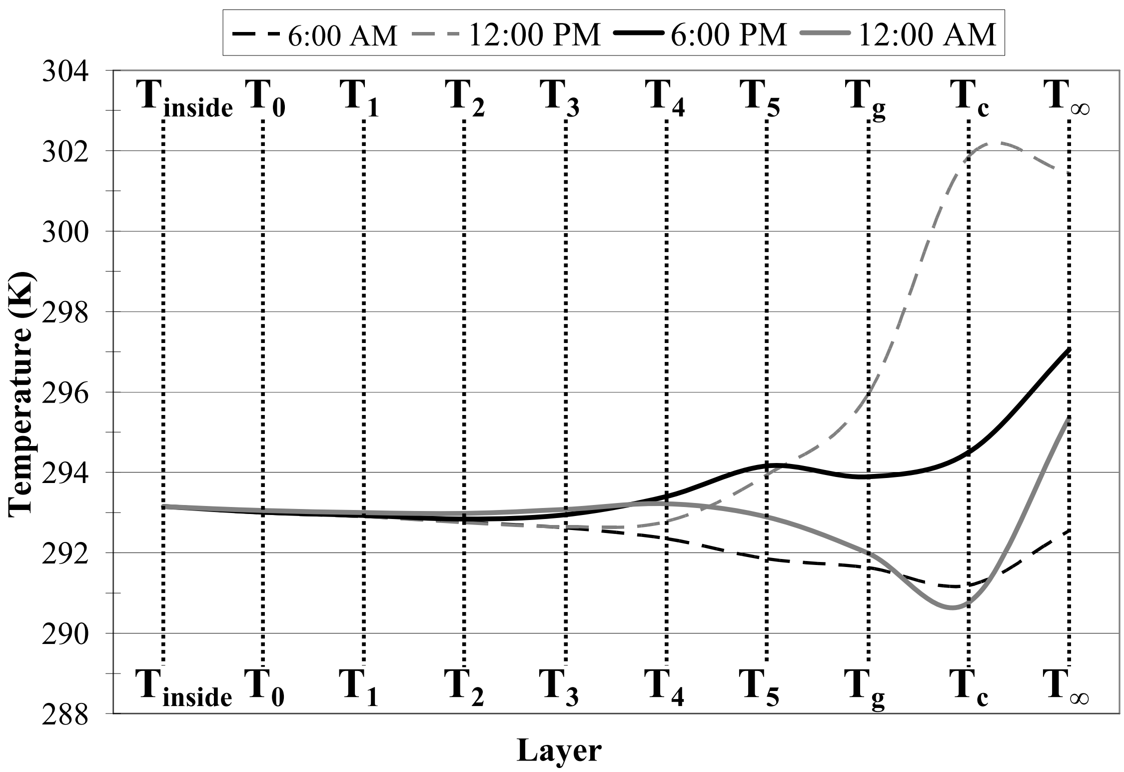


Figure 17: Temperature profile on JD 198 for NY simulation.

Moving toward the canopy layer (T_c), the investigated times can be ranked according to decreasing temperature, 12:00 PM >> 6:00 PM > 6:00 AM > 12:00AM, with the 12:00PM temperature being much greater than any other canopy temperature, by at least

8 K. These results are justified due to the presence of solar radiation and increased ambient temperatures, shown in the graph by the T_∞ layer.

The CA temperature profile has approximately the same temperature until layer T_3, however unlike the NY simulation, these similar values steadily decrease from the constant inside temperature value. This is due to the fact that for all the investigated times (with the exception of 12:00 PM) $T_\infty < T_{inside}$, shown below in Figure 18.

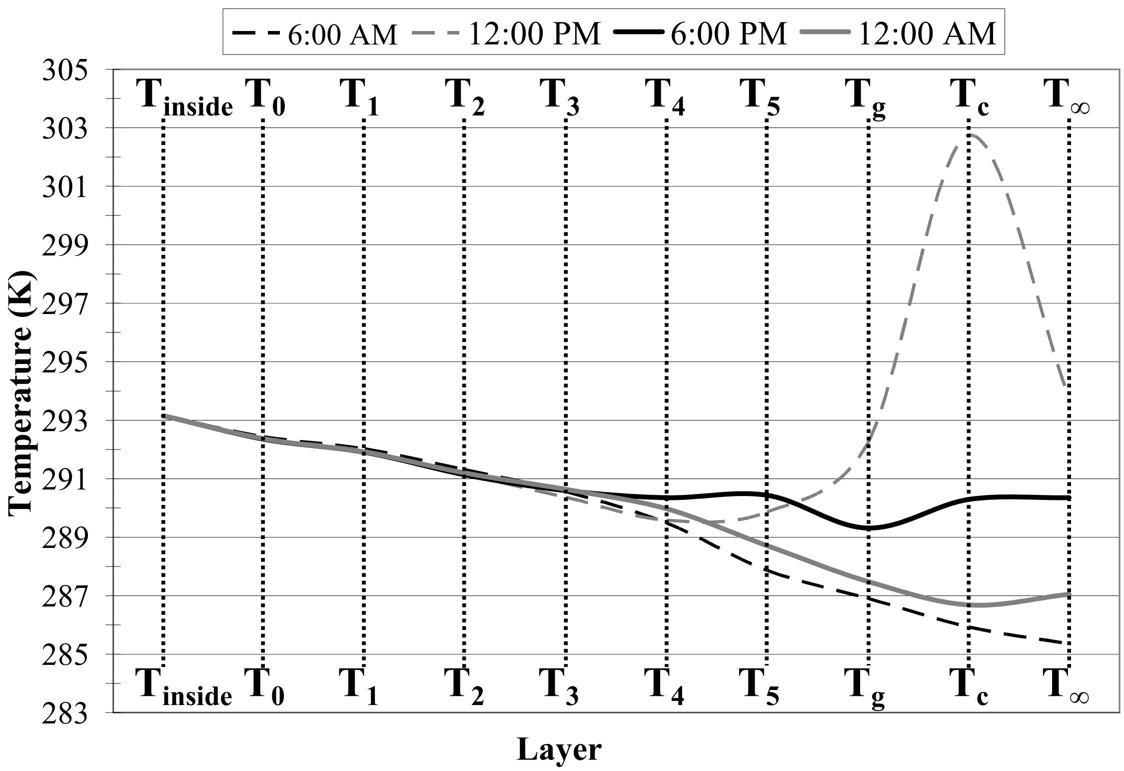


Figure 18: Temperature profile on JD 198 for CA simulation.

The investigated times can again be ranked according to decreasing temperature at the canopy layer, 12:00 PM >> 6:00 PM > 12:00 AM > 6:00AM, with the 12:00PM temperature being much greater than any other canopy temperature, by at least 12 K.

The AZ temperature profile also has approximately the same temperature until layer T_3; however these similar values steadily increase from the constant inside temperature value. This is due to the fact that for all the investigated times $T_\infty >> T_{inside}$, shown below in Figure 19.

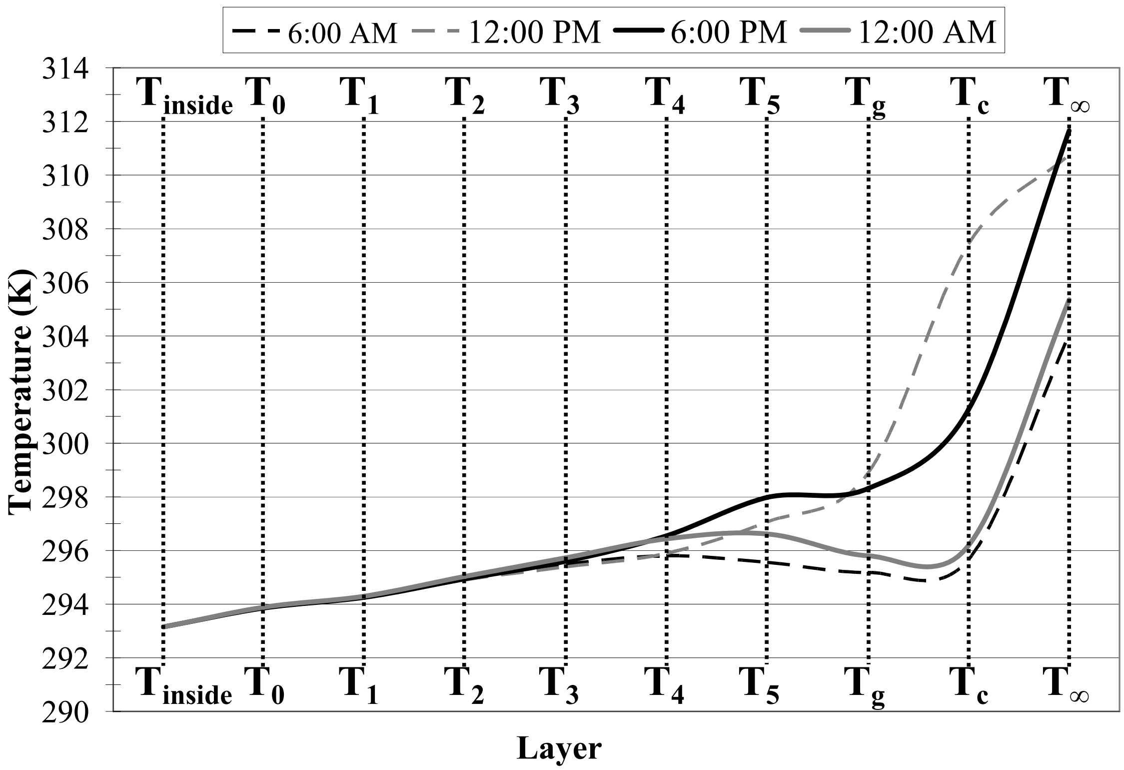


Figure 19: Temperature profile on JD 198 for AZ simulation.

The investigated times can once again be ranked according to decreasing temperature at the canopy layer, 12:00 PM >> 6:00 PM > 12:00 AM > 6:00AM, with the 12:00PM temperature being much greater than any other canopy temperature, by 7 K for the 6:00 PM time and 12 K for they AM times.

Flux Analysis

The flux analysis calculated the flux at the T_0 layer (boundary between the inside and the support structure) to compare the performance of a green roof to a non-green roof in each of the modeled locations. The green roof calculations used the base data as previously described, while the non-green roof calculations assumed that $LAI = 0$, $\varepsilon_c = 0$, and $g_v = 0$. The MATLAB program was run for each of the three locations using these new constraints that represent no plants on the roof. Note that in Equation 12 (the energy balance for the canopy layer) all terms drop out except the convection between the layer and the ambient air, which is expected for a layer with no plants or thermal storage.

The only term of interest in the flux analysis is the flux out of the T_0 layer, which represents the flow of energy from the outside to the inside (a positive value). A negative value of flux out means the flow of energy is from the inside to the outside. From Equation 1, the flux out is the convection term between the T_0 layer and the inside air, or $h_{inside}(T_0 - T_{inside})$. Figure 20 below shows a schematic of the T_0 layer being investigated including this flux out term.

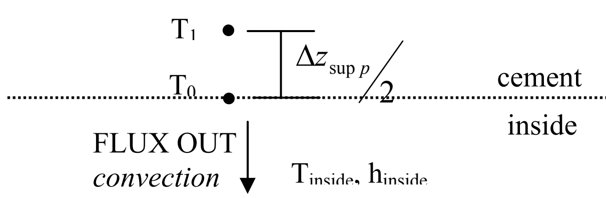


Figure 20: Schematic of the T_0 layer including FLUX OUT term.

For each of the modeled locations, the green roof flux out and the non-green roof flux out were plotted to investigate the differences between them. The integral, or area underneath the curves, was calculated for each roof condition over the summer months, June 21 – September 22 (JD 172-265), because a negative flux (heat loss from the building) represents a lower cooling load, and a reduced need for air conditioning, in the summer. A full year long comparison is not possible since the green roof model does not take into account the operation of the entire building. A building design and control model, which was not within the scope of this study, is needed to compare the yearly performance. In addition to this seasonal comparison, the same day used for the temperature profile (JD 198 – July 17) was chosen to represent the average cooling loads for a day. Figures 21-23 below show the flux out plots of the NY, CA, and AZ simulations respectively. The shaded areas represent the summer months that were further investigated.

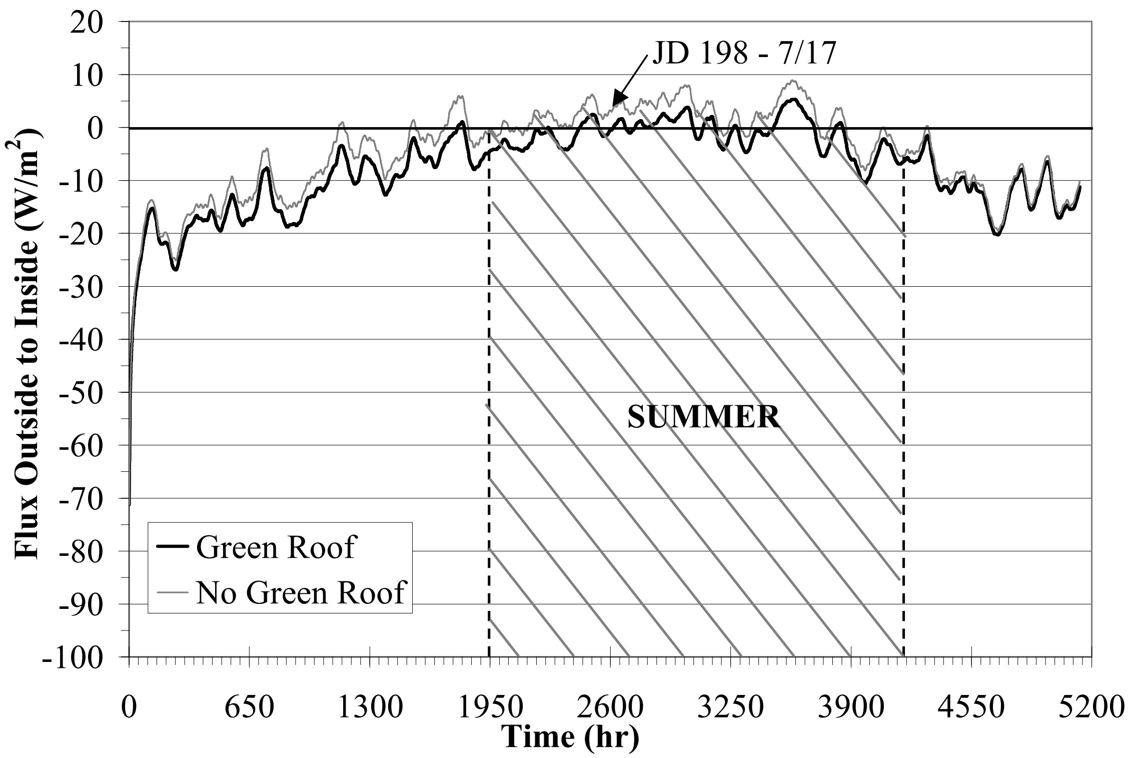


Figure 21: Flux out analysis for the NY simulation comparing a green roof to a non-green roof.

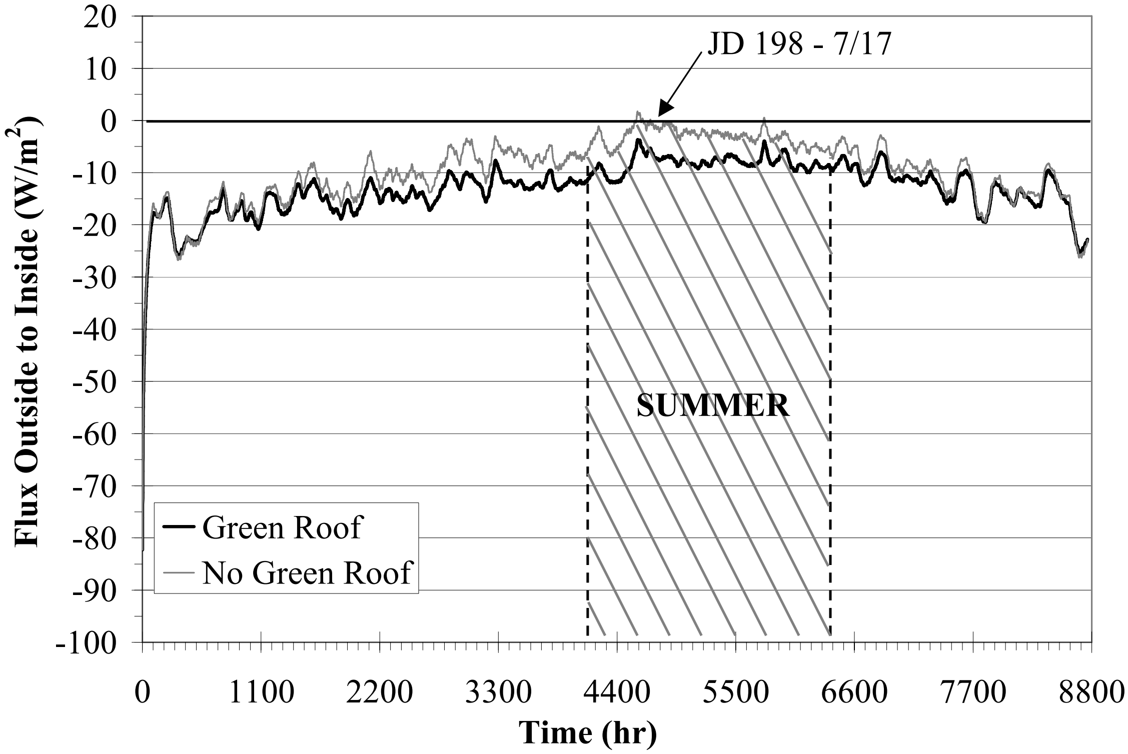


Figure 22: Flux out analysis for the CA simulation comparing a green roof to a non-green roof.

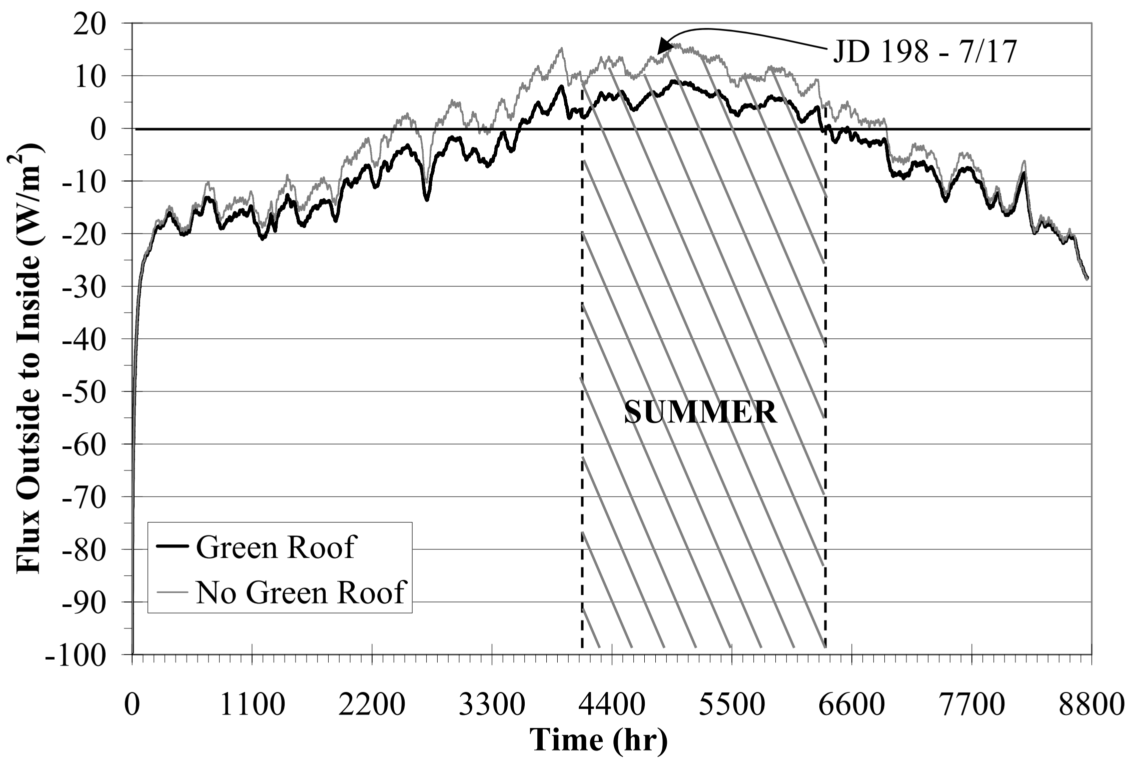


Figure 23: Flux out analysis for the AZ simulation comparing a green roof to a non-green roof.

The positive flux out values during the summer months in Figures 21 and 23 do not represent a reduced need for cooling. Since the flux is positive, the energy flows from the outside to the inside, indicating an increased need in air conditioning to maintain the constant indoor air temperature. However, comparisons can still be made on overall performance since for all the locations the green roof flux out curves are consistently a lesser value than the non-green roof flux out. Table 6 below provides the results of the seasonal and snapshot flux analysis with and without a green roof.

Table 6: Flux out analysis results for the three modeled locations.

	Summer		JD 198 (July 17)	
Location	**Green Roof**	**No Green Roof**	**Green Roof**	**No Green Roof**
New York, NY	-3.14 kW/m^2	4.21 kW/m^2	-32 W/m^2	59 W/m^2
Santa Maria, CA	-18.45 kW/m^2	-6.86 kW/m^2	-164 W/m^2	-27 W/m^2
Phoenix, AZ	11.76 kW/m^2	25.30 kW/m^2	154 W/m^2	310 W/m^2

The reduced amount of energy a green roof provides can be calculated by taking the absolute value of the difference between the green roof and non-green roof values, or | Green Roof – No Green Roof |. Table 7 below provides the amount of energy that is saved by installing a green roof in each of the modeled locations. Note that all locations see a benefit from a green roof, with a green roof in AZ having the greatest impact.

Table 7: Amount of energy saved by installing a green roof.

Location	Summer	JD 198 (July 17)
New York, NY	7.4 kW/m^2	91 W/m^2
Santa Maria, CA	11.6 kW/m^2	137 W/m^2
Phoenix, AZ	13.5 kW/m^2	156 W/m^2

Conclusion

Previously conducted research on extensive green roof systems mostly focused on the plant science aspects, the hydrological benefits of installation, and observations on thermal performance. There is currently a lack of quantifiable data and an energy model for green roofs. This study aims to rectify this problem by creating an energy model and subsequent computer program to calculate the temperature profiles and ultimately the energy savings that a green roof provides.

The energy model was developed using a lumped parameter approach with one dimensional, time dependent heat transfer. The green roof was broken up into three sections, support structure, soil, and canopy, consisting of a total of eight layers. Energy balances for each layer were developed based on the general formula of FLUX IN – FLUX OUT = STORAGE. Since the eight energy balances must be solved simultaneously, an 8x8 matrix equation had to be set up. To solve this matrix equation, a computer program was written in the MATLAB language that outputs a text file which contains hourly temperatures for each of the eight layers. Three locations with different climates were modeled: New York, NY (temperate), Santa Maria, CA (Mediterranean), and Phoenix, AZ (desert). Weather data for each of these locations was taken from the ASHRAE Technical Committee 4.2: Weather Information CD-ROM.

The created model has to be validated prior to running any simulations to prove that reasonable results were computed. This was accomplished by supplying the computer program with a sample 24 hour weather data set which was run for ten consecutive days. Each layer eventually approached a steady value or a diurnal cycle with both outcomes validating the model.

A sensitivity analysis was performed by allowing six parameters to vary: thermal conductivity of the structural support, reflectivity of the soil, LAI, depth of the soil layer, thermal conductivity of the soil, and thermal diffusivity of the soil. Changing the values of reflectivity of the soil and thermal diffusivity of the soil produced no significant differences between simulations. Varying the thermal conductivity of the structural support, or whether insulation was or was not present in the roof, showed that insulation only had an impact during cold periods for the NY and AZ simulations in the layers closest to the insulation. However, for the CA simulations, there was a noticeable difference between an insulated roof and a non-insulated roof throughout the year, due to the locations Mediterranean climate. Varying the LAI did not have a large affect on any of the simulations, however for the first seven layers, the minimum value of LAI had the highest temperature. A smaller LAI should result in a higher temperature since more solar radiation is hitting the soil surface and less shading is being provided by the plants. The top layer of the model exhibits an opposite relationship as the maximum value of LAI has the highest temperature, due to the increased area for radiation and moisture exchange. Varying the depth of the soil layer showed that increasing the amount of soil led to less temperature variability and greater thermal stability, since a larger thermal mass was created with the additional soil. Finally, for the thermal conductivity of the soil, the maximum value resulted in the lowest temperature, since a larger thermal conductivity results in more conduction occurring, therefore reducing the temperature.

Temperature profiles were generated for each location in six hour intervals (12:00 AM, 6:00 AM, 12:00 PM, and 6:00 PM) on July 17 (JD 198). For all the locations the temperature of the canopy layer (T_c) at 12:00 PM was much greater than at any other time. This is due to the presence of solar radiation and increased ambient air temperatures (T_∞).

A flux analysis was conducted on the T_0 layer to quantify the amount of energy saved by utilizing green roof technology. It was found that during the summer months a green roof in NY saved 7.4 kW/m^2 of energy, compared to one in CA which saved 11.6 kW/m^2 of energy. The greatest energy savings were found in AZ, where during the summer 13.5 kW/m^2 of energy was saved. For a daily snapshot, JD 198 was once again used and it was found for NY a green roof saved 91 W/m^2, compared to one in CA which saved 137 W/m^2. Once again the greatest energy savings were found in AZ, where a green roof was found to save 156 W/m^2 of energy.

Other simulations can be run on this green roof model and their results can be quantified in a similar manner. The modeler has complete flexibility in changing the parameters to make them applicable to their situation and location. A complete energy balance for a house with a green roof can be conducted using this model, along with a building design and control model to fully compute the yearly performance. The results presented are only the beginning of the simulations and investigations that can be conducted using this model.

Acknowledgements

This project would not be completed without the continued support and advice of Dr. Louis Albright. Without his help, many of the obstacles along the way would not have been successfully tackled. Dr. Albright always made himself available to me and for that I am extremely grateful. Another thank you to Brenda Marchewka, who saved me countless hours figuring out and dealing with the administrative side of things. Thank you to Akio Enders for providing me with much of the literature to get me on track for the project and Dr. Thomas Whitlow who explained the plant science workings of green roofs to a very inquisitive engineer. Finally a big thank you to my family for their continued support throughout this entire process. I would not be where I am today without them.

References

Albright, Louis D. Environment Control for Animals and Plants. St. Joseph, MI: American Society of Agricultural and Biological Engineers, 1990.

Bakker, J.C., et al. Greenhouse Climate Control: an integrated approach. Wageningen, The Netherlands: Wageningen Pers, 1995.

Bass, Brad. "Greening of Cities." Encyclopedia of Global Environmental Change 3 (2001): 356-362.

Campbell, Gaylon S., and John M. Norman. An Introduction to Environmental Biophysics. New York: Springer-Verlag, 1998.

Del Barrio, Elena Palomo. "Analysis of the green roofs cooling potential in buildings." Energy and Buildings 27 (1998): 179-193.

Dunnett, Nigel and Alison Nolan. "The Effect of Substrate Depth and Supplementary Watering on the Growth of Nine Herbaceous Perennials in a Semi-Extensive Green Roof." Acta Horticulturae (ISHS) 643 (2004): 305-309.

Hanks, John, and J.T. Ritchie. Modeling Plant and Soil Systems. Madison, WI: American Society of Agronomy, Inc.: Crop Science Society of America, Inc.: Soil Science Society of America, Inc., 1991.

Leckie, Jim, et al. <u>More Other Homes and Garbage: Designs for Self-sufficient Living</u>. San Francisco: Sierra Club Books, 1981.

Liu, Karen and Bas Bakaran. “Thermal Performance of Green Roofs Through Field Evaluation.” <u>Proceedings for the First North American Green Roof Infrastructure Conference, Awards, and Trade Show, Chicago, IL., May 29-30, 2003</u>. Ontario, Canada: Green Roofs for Health Cities, 2003. 1-10.

Rowe, Bradley. “Green Roofs and the Ford Rouge Project.” <u>Combined Proceedings International Plant Propagators’ Society</u> 52 (2002): 492-495.

Thompson, J. William. “Grass-Roof Movements.” <u>Landscape Architecture</u> 88.5 (1998): 47-51.

Wark, Christopher G., and Wendy W. Wark. “Green Roof: Specifications and Standards.” <u>The Construction Specifier</u> 56.8 (2003): 76-82.

Young, Hugh D. <u>University Physics</u>. San Francisco: Pearson Addison Wesley, 2004.

Appendix A

List of Symbols

Symbol	Units	Description
CC		*fraction of sky covered with cloud*
cp_{soil}	{ J/kg }	*specific heat of soil*
cp_{supp}	{ J/kg }	*specific heat of structural support*
e^{-kLAI}		*fraction of solar radiation transmitted through the canopy*
g_v	{ mol m^{-2} hr^{-1} }	*conductance for vapor*
h_{air}	{ J m^{-2} hr^{-1} K^{-1} }	*convection coefficient of air*
h_{inside}	{ J m^{-2} hr^{-1} K^{-1} }	*convection coefficient of inside air*
k		*canopy extinction coefficient*
k_{soil}	{ J m^{-1} hr^{-1} K^{-1} }	*thermal conductivity of soil*
k_{supp}	{ J m^{-1} hr^{-1} K^{-1} }	*thermal conductivity of structural support*
LAI		*Leaf Area Index*
P	{ Pa }	*standard atmospheric pressure*
pw	{ Pa }	*partial vapor pressure*
pws	{ Pa }	*saturated vapor pressure*
$pws(T_c)$	{ Pa }	*saturated vapor pressure at canopy temperature*
$pws(T_g)$	{ Pa }	*saturated vapor pressure at ground temperature*
$RAD_{solar,\ in,\ g}$	{ J m^{-2} hr^{-1} }	*solar radiation in to the ground*
$RAD_{thermal,\ out,\ g \to c}$	{ J m^{-2} hr^{-1} }	*thermal radiation out between ground/canopy*
$RAD_{thermal,\ out,\ g \to sky}$	{ J m^{-2} hr^{-1} }	*thermal radiation out between ground/sky*
rH		*relative humidity*
T_{ave}^{3}	{ K^3 }	*cubed average temperatures*
T_c	{ K }	*canopy temperature*
t_d	{ K }	*dew point temperature*
T_g	{ K }	*ground temperature*
T_{inside}	{ K }	*inside air temperature*
T_{sky}	{ K }	*sky temperature*
T_∞	{ K }	*ambient air temperature*
T_0	{ K }	*temperature of boundary between inside air/structural support*
T_1	{ K }	*temperature of structural support*
T_2	{ K }	*temperature of soil layer #1*
T_3	{ K }	*temperature of soil layer #2*
T_4	{ K }	*temperature of soil layer #3*
T_5	{ K }	*temperature of soil layer #4*
$T_{1,t-1}$	{ K }	*temperature of structural support from previous time step*

$T_{2,t-1}$	{ K }	*temperature of soil layer #1 from previous time step*
$T_{3,t-1}$	{ K }	*temperature of soil layer #2 from previous time step*
$T_{4,t-1}$	{ K }	*temperature of soil layer #3 from previous time step*
$T_{5,t-1}$	{ K }	*temperature of soil layer #4 from previous time step*
u	{ m/s }	*wind speed*
$Vapor\ Loss_g$	{ J m^{-2} hr^{-1} }	*vapor loss from the ground*

Greek

α_{soil}		*absorptivity of ground*
α_2	{ m^2/hr }	*thermal diffusivity of the soil*
ΔT	{ K }	*change in temperature*
Δz_{soil}	{ m }	*depth of soil layer*
Δz_{supp}	{ m }	*depth of structural support*
ε		*emissivity*
ε_c		*emissivity of canopy*
ε_g		*emissivity of ground*
λ	{ J/mol }	*latent heat of vaporization of water*
ρ_c		*reflectivity of canopy*
ρ_{soil}	{ kg/m^3 }	*density of soil*
ρ_{supp}	{ kg/m^3 }	*density of structural support*
σ	{ J m^{-2} hr^{-1} K^{-4} }	*Stefan-Boltzmann constant*
ϕ_s	{ J/m^2 }	*global horizontal irradiance*

Appendix B

Parameter Values

cp_{supp} =	800 J/kg
h_{inside} =	32400 J/m^{2}hrK
k =	0.74
P =	101,325 Pa
T_{inside} =	293.15 K
Δz_{supp} =	0.1 m
ε_c =	0.95
ε_g =	0.9
λ =	44000 J/mol
ρ_c =	0.25
ρ_{supp} =	2400 kg/m^3
σ =	0.000204 J/m^{2}hrK^4

Varying Parameters

e^{-kLAI} =	0.1086; when $LAI = 3$	
	0.4771; when $LAI = 1$	
	0.0247; when $LAI = 5$	
k_{soil} =	5400 J/mhrK	(BASE)
	2880 J/mhrK	(MIN)
	7200 J/mhrK	(MAX)
k_{supp} =	2880 J/mhrK	(no insulation)
	288 J/mhrK	(insulation [k_{supp}/10])
LAI =	3	(BASE)
	1	(MIN)
	5	(MAX)
α_{soil} =	0.92; when $\rho_g = 0.08$	
	0.65; when $\rho_g = 0.35$	
	0.82; when $\rho_g = 0.18$	
α_2 =	0.00216 m^2/hr	(BASE)
	0.00144 m^2/hr	(MIN)
	0.00288 m^2/hr	(MAX)
Δz_{soil} =	0.5 m	(BASE)
	0.25 m	(MIN)
	1 m	(MAX)
ρ_g =	0.08; soil, wet dark	(BASE)
	0.35; sand, dry white	(MAX)
	0.18; soil, dry light	(MID)

Appendix C

Weather Files

AZphoenixweather.txt

Hour	Global Horizontal Irradiance (J/m^2)	Dry Bulb Temperature (K)	Wind Speed (m/s)	Opaque Sky Cover	Relative Humidity
1	0	274.25	1.5	0.0	0.77
2	0	274.25	1.5	0.0	0.77
3	0	274.05	2.0	0.0	0.76
4	0	273.95	2.6	0.0	0.75
5	0	273.75	3.1	0.0	0.73
6	0	273.35	2.4	0.0	0.76
7	0	273.05	1.7	0.0	0.77
8	26000	272.55	1.0	0.0	0.80
9	404000	275.55	1.5	0.0	0.66
10	1031000	278.45	2.1	0.0	0.54
11	1562000	281.45	2.6	0.0	0.44
12	1914000	282.95	2.2	0.0	0.40
13	2047000	284.45	1.9	0.0	0.37
14	1944000	285.95	1.5	0.0	0.33
15	1606000	286.35	1.0	0.0	0.31
16	1088000	286.65	0.5	0.0	0.30
17	485000	287.05	0.0	0.0	0.28
18	42000	284.85	0.5	0.0	0.34
19	0	282.55	1.0	0.0	0.43
20	0	280.35	1.5	0.0	0.53
21	0	279.25	2.0	0.0	0.60
22	0	278.15	2.6	0.0	0.68
23	0	277.05	3.1	0.0	0.76
24	0	276.45	2.9	0.0	0.78

CAsantamariaweather.txt

Hour	Global Horizontal Irradiance (J/m^2)	Dry Bulb Temperature (K)	Wind Speed (m/s)	Opaque Sky Cover	Relative Humidity
1	0	277.55	1.0	0.7	0.79
2	0	278.15	1.5	0.8	0.73
3	0	277.55	1.5	1.0	0.74
4	0	277.55	1.5	0.9	0.76
5	0	277.55	0.0	0.9	0.76
6	0	277.05	1.0	0.8	0.76
7	0	277.05	1.0	0.8	0.70
8	55000	277.05	1.0	0.5	0.76
9	316000	278.75	1.5	0.3	0.70
10	1111000	285.35	2.1	0.0	0.45
11	1580000	289.85	0.0	0.0	0.27
12	1762000	292.05	1.5	0.1	0.30
13	1676000	293.15	3.6	0.2	0.20
14	970000	290.95	5.1	0.4	0.30
15	533000	289.85	3.6	1.0	0.38

CAsantamariaweather.txt continued

Hour	Global Horizontal Irradiance (J/m^2)	Dry Bulb Temperature (K)	Wind Speed (m/s)	Opaque Sky Cover	Relative Humidity
16	225000	289.25	2.1	1.0	0.44
17	45000	287.05	1.5	1.0	0.39
18	0	285.95	2.1	1.0	0.47
19	0	284.25	1.0	1.0	0.56
20	0	284.85	1.5	1.0	0.50
21	0	283.75	1.0	1.0	0.50
22	0	283.75	0.0	1.0	0.50
23	0	283.75	0.0	1.0	0.46
24	0	283.75	1.5	1.0	0.46

NYCweather.txt

Hour	Global Horizontal Irradiance (J/m^2)	Dry Bulb Temperature (K)	Wind Speed (m/s)	Opaque Sky Cover	Relative Humidity
1	0	279.65	6.5	1.0	0.66
2	0	279.35	7.2	1.0	0.66
3	0	279.15	7.6	1.0	0.66
4	0	279.05	7.9	1.0	0.66
5	0	279.15	7.9	1.0	0.66
6	0	279.35	7.6	1.0	0.65
7	190000	279.25	6.7	1.0	0.65
8	485000	280.15	5.8	1.0	0.61
9	1485000	281.15	5.0	1.0	0.58
10	2013000	282.05	4.1	0.8	0.56
11	2101000	282.95	5.1	0.8	0.54
12	2337000	283.95	6.2	0.6	0.52
13	2724000	284.85	7.2	0.3	0.50
14	2242000	284.45	7.0	0.6	0.53
15	863000	284.15	6.9	0.8	0.57
16	578000	283.75	6.7	1.0	0.61
17	441000	282.95	5.8	1.0	0.65
18	117000	282.25	5.0	0.8	0.69
19	4000	281.45	4.1	0.8	0.74
20	0	281.85	3.4	0.7	0.70
21	0	282.15	2.8	0.7	0.67
22	0	282.55	2.1	0.6	0.63
23	0	281.85	2.1	0.6	0.67
24	0	281.25	2.1	0.7	0.71

Appendix D

Program and Supporting Functions

```
clear
%   Script file GreenRoofModel.m
%   Purpose:
%       This program will allow the user to input which location in the
%   US that is to be modeled.  Weather data will be imported based on
%   input entry.  A series of functions are called which place values
%   into the green roof heat transfer model matrix.  The matrix is
%   solved for the temperatures of each layer.
%            Copyright 2006 William Striar Lambert
%   File:       GreenRoofModel.m
%   Name:       William Lambert
%   Date:       December 5, 2005
%   Modified:   December 6, 2005
%               December 7, 2005
%               December 14, 2005
%               December 19, 2005
%               December 20, 2005
%               December 21, 2005
%               January 25, 2006
%               January 26, 2006
%               January 28, 2006
%               February 6, 2006
%               February 16, 2006
%               February 23, 2006
%               February 24, 2006
%
%   Variable Dictionary:
%
%   vap_cond:   conductance for vapor (mol m^-2 hr^-1):
%               (From Campbell 234)
%               Equation:  (0.12 * u)/(0.6 + 0.2 * u);
%               u = wind speed (m/s)
%   LAT_VAP:    latent heat of vaporization of water (J/mol)
%               (From CB 37) (global)
%   PRESSURE:   atmospheric pressure at STP (Pa) (global)
%   RHO_C:      reflectivity of canopy (grass) (From CB 172) (global)
%   solar_in:   global horizontal irradiance array (From data sets)
%               (J/m^2)
%   ext_coeff:  extinction coefficient
%               (From Greenhouse Climate Control Ch2)
%   EPSILON_C:  emissivity of canopy (global)
%   SIGMA:      Stefan-Boltzmann constant (J m^-2 hr^-1 K^-4) (global)
%   EPSILON_G:  emissivity of ground (global)
%   z_supp:     support depth (m)
%   dens_supp:  density of support (concrete) (kg/m^3)
%   cp_supp:    specific heat of support (concrete) (J/kg)
%   ABS_G:      absorptivity of ground:  1 - RHO_G  (global)
%   H_AIR:      convective heat transfer coefficient to air
%               (J/m^2 hr K) (global)
%   h_inside:   convective heat transfer coefficient to inside
%               (J/m^2 hr K) (From "Better Homes and Garbage" P. 153)
```

```
%   e_kLAI:     exp(-ext_coeff * LAI)  (global)
%   user_loc:   user input of modeling location from menu command
%   jul_date:   julian date array (from data sets)
%   hour:       hour array (from data sets)
%   air_temp:   air temperature array (from data sets) (K)
%   wind:       wind speed array (from data sets) (m/s)
%   cloud_cov:  cloud cover array (from data sets)
%   rel_hum:    relative humidity array (from data sets) (percent)
%   date:       day counter for calculation loop
%   hourly:     hourly counter for calculation loop
%   hour_count: hourly counter (cumulative) for matrix definition
%   irr:        solar_in value for specific hour (J/m^2)
%   t_air:      air_temp value for specific hour (K)
%   vel:        wind value for specific hour (m/s)
%   cloudy:     cloud_cov value for specific hour
%   rH:         rel_hum value for specific hour
%   T_sky:      sky temperature (K)
%   avgTgTc:    cube of the average ground and canopy temperatures
%                   avgTgTc = (( Tg + Tc ) / 2) ^ 3 (K^3)
%   avgTgTsky:  cube of the average ground and sky temperatures
%                   avgTgTsky = (( Tg + T_sky ) / 2) ^ 3 (K^3)
%   avgTcTsky:  cube of the average canopy and sky temperatures
%                   avgTcTsky = (( Tc + T_sky ) / 2) ^ 3 (K^3)
%   Tg:         ground temperature (K)
%   Tc:         canopy temperature (K)
%   T_inside:   inside air temperature (K)
%   T0t1:       previous hours temp. for inside/cement boundary balance
%                (K)
%   T1t1:       previous hours temperature for cement balance (K)
%   T2t1:       previous hours temperature for soil #1 balance (K)
%   T3t1:       previous hours temperature for soil #2 balance (K)
%   T4t1:       previous hours temperature for soil #3 balance (K)
%   T5t1:       previous hours temperature for soil #4 balance (K)
%   Tgt1:       previous hours temp. for ground/canopy boundary balance
%                (K)
%   Tct1:       previous hours temperature for canopy/air boundary
%               balance (K)
%   var_mat:    8x8 coefficient matrix
%   loopvar:    counter for linearization loop
%   soil_Tg_term: function that calculates the coefficient for
%                 var_mat(7,7)
%   soil_Tc_term: function that calculates the coefficient for
%                 var_mat(7,8)
%   canopy_Tg_term: function that calculates the coeff. for
%                   var_mat(8,7)
%   canopy_Tc_term: function that calculates the coeff. for
%                   var_mat(8,8)
%   sol_mat:    8x1 solution matrix
%   data:       temporary storage array for output to functions
%   soil_matrix_ans: function that calculates the value for
%                    sol_mat(7,1)
%   canopy_matrix_ans:  function that calculates the value for
%                       sol_mat(8,1)
%   temp_mat:   8x1 temperature matrix [T0 T1 ..... Tc]
%
%   VARIED FOR SENSITIVITY ANALYSIS
%   --------------------------------
```

```
%   k_supp:     thermal conductivity of support (concrete)
%               (J m^-1 hr^-1 K^-1) (From "Unversity Physics" Young)
%   RHO_G:      reflectivity of ground (From CB 172) (global)
%   LAI:        leaf area index (global)
%   Z_G:        soil depth (m) (global)
%   K_G:        thermal conductivity of ground (J m^-1 hr^-1 K^-1)
%               (global)(From "Modeling plant and soil systems" p. 405)
%   ALPHA_2:    thermal diffusivity of ground (m^2/hr)  (global)
%               (From "Modeling plant and soil systems" p. 405)

    global RHO_G e_kLAI Z_G K_G ALPHA_2 LAT_VAP PRESSURE ...
        RHO_C EPSILON_C SIGMA EPSILON_G ABS_G ...
        H_AIR LAI

%   VARIABLE DEFINITION
%=======================================================================
%   VARYING PARAMETERS

    k_supp = (0.8) * 3600;       % no insulation (J m^-1 hr^-1 K^-1)
%   k_supp = (0.08) * 3600;      % insulation (k/10)(J m^-1 hr^-1 K^-1)

    RHO_G = 0.08;                % BASE - soil, wet dark
%   RHO_G = 0.35;                % MAX - sand, dry white
%   RHO_G = 0.18;                % MID - soil, dry light

    LAI = 3;                     % BASE
%   LAI = 1;                     % MIN
%   LAI = 5;                     % MAX

    Z_G = 0.5;                   % BASE - m
%   Z_G = 0.25;                  % MIN - m
%   Z_G = 1;                     % MAX - m

    K_G = (1.5) * 3600;          % BASE - (J m^-1 hr^-1 K^-1)
%   K_G = (0.8) * 3600;          % MIN - (J m^-1 hr^-1 K^-1)
%   K_G = (2) * 3600;            % MAX - (J m^-1 hr^-1 K^-1)

    ALPHA_2 = (0.6e-6) * 3600;   % BASE - m^2/hr
%   ALPHA_2 = (0.4e-6) * 3600;   % MIN - m^2/hr
%   ALPHA_2 = (0.8e-6) * 3600;   % MAX - m^2/hr
%   -------------------------------------------------------------------

    LAT_VAP = 44000;             % J/mol
    PRESSURE = 101325;           % Pa
    RHO_C = 0.25;
    ext_coeff = 0.74;
    EPSILON_C = 0.95;
    SIGMA = (5.67e-8) * 3600;    % J m^-2 hr^-1 K^-4
    EPSILON_G = 0.9;
    z_supp = 0.1;                % m
    dens_supp = 2400;            % kg/m^3
    cp_supp = 800;               % J/kg
    ABS_G = 1 - RHO_G;
    h_inside = (9) * 3600;       % (J m^-2 hr^-1 K^-1)
    T_inside = 293.15;           % K
```

```
%   INITIAL CONDITIONS

    avgTgTc = 279.65;            % K^3
    avgTgTsky = 279.65;          % K^3
    avgTcTsky = 279.65;          % K^3
    T_sky = 279.65;              % K
    Tg = 279.65;                 % K
    Tc = 279.65;                 % K
%======================================================================

%   CALCULATIONS

    e_kLAI = exp(-ext_coeff * LAI);
    avgTgTc = ((Tg + Tc)/2)^3;
    avgTgTsky = ((Tg + T_sky)/2)^3;
    avgTcTsky = ((Tc + T_sky)/2)^3;

%   USER SELECTION OF LOCATION

    user_loc = menu('Choose a modeling location', 'NYC, NY', ...
        'Phoenix, AZ', 'Santa Maria, CA');

%   IMPORT OF TEXT FILES

   switch (user_loc)
   case 1,
       jul_date = 91:304;
       hour = zeros(1, 5136);
       solar_in = zeros(1, 5136);
       air_temp = zeros(1, 5136);
       wind = zeros(1, 5136);
       cloud_cov = zeros(1, 5136);
       rel_hum = zeros(1, 5136);

       [hour, solar_in, air_temp, wind, cloud_cov, rel_hum]= ...
           textread('NYCweather.txt', '%d %f %f %f %f %f');
   case 2,
       jul_date = 1:365;
       hour = zeros(1, 8760);
       solar_in = zeros(1, 8760);
       air_temp = zeros(1, 8760);
       wind = zeros(1, 8760);
       cloud_cov = zeros(1, 8760);
       rel_hum = zeros(1, 8760);

       [hour, solar_in, air_temp, wind, cloud_cov, rel_hum]= ...
           textread('AZphoenixweather.txt', '%d %f %f %f %f %f');
   case 3,
       jul_date = 1:365;
       hour = zeros(1, 8760);
       solar_in = zeros(1, 8760);
       air_temp = zeros(1, 8760);
       wind = zeros(1, 8760);
       cloud_cov = zeros(1, 8760);
```

```
        rel_hum = zeros(1, 8760);

        [hour, solar_in, air_temp, wind, cloud_cov, rel_hum]= ...
            textread('CAsantamariaweather.txt', '%d %f %f %f %f %f');
    otherwise,
    end

%   Initializing previous temp values in solution matrix
    T1t1 = air_temp(1);
    T2t1 = air_temp(1);
    T3t1 = air_temp(1);
    T4t1 = air_temp(1);
    T5t1 = air_temp(1);

%   MATRIX CALCULATIONS

    date = 0;
    hourly = 0;
    hour_count = 0;

    fid = fopen('TempOut.txt', 'wt');

    for date = 1:length(jul_date)           % loop for each trial day

%         fprintf(fid, '%6d\n', jul_date(date));

        for hourly = 1:24                   % loop for 24 hrs each day
            hour_count = hour_count + 1;    % cumulative hourly counter

            %  extraction of specific hourly data from array
            irr = solar_in(hour_count);
            t_air = air_temp(hour_count);
            vel = wind(hour_count);
            cloudy = cloud_cov(hour_count);
            rH = rel_hum(hour_count);

            % Calculation of vapor conductance (mol m^-2 hr^-1)
            vap_cond = ((0.12*vel)/(0.6 + (0.2 * vel))) * 3600;

            % Calculation of Tsky (K)
            T_sky = t_air - ((t_air - (0.0552 * (t_air)^1.5)) ...
                        * (1 - cloudy));

            % Calculation of H_AIR (J m^-2 hr^-1 K^-1)
            H_AIR = (11.67 + (3.33 * vel)) * 3600;
            % Units: 11.67 W m^-2 K^-1; 3.33 J m^-3 K^-1

            % Placement of values in variable matrix
            var_mat = zeros(8, 8);

            for loopvar = 1:3

                % Row 1 (inside)
                var_mat(1,1) = k_supp/(z_supp/2) + h_inside;
```

```
var_mat(1,2) = -k_supp/(z_supp/2);

% Row 2 (cement)
var_mat(2,1) = -k_supp/(z_supp/2);
var_mat(2,2)= ((8 * k_supp * K_G)/((k_supp * Z_G) ...
                 + (4 * K_G * z_supp))) ...
                 + (k_supp/(z_supp/2)) ...
                 + (dens_supp * cp_supp * z_supp);
var_mat(2,3) = -(8 * k_supp * K_G)/((k_supp * Z_G) ...
                 + (4 * K_G * z_supp));

% Row 3 (soil layer 1)
var_mat(3,2) = -(8 * k_supp * K_G)/((k_supp * Z_G) ...
                 +(4 * K_G * z_supp));
var_mat(3,3) = ((8 * k_supp * K_G)/((k_supp * Z_G) ...
                 + (4 * K_G * z_supp))) ...
                 + (K_G/(Z_G/4)) ...
                 + (((K_G/ALPHA_2) * (Z_G/4)));
var_mat(3,4) = -K_G/(Z_G/4);

% Row 4 (soil layer 2)
var_mat(4,3) = -K_G/(Z_G/4);
var_mat(4,4) = ((2 * K_G)/(Z_G/4)) ...
              + ((K_G/ALPHA_2) * (Z_G/4));
var_mat(4,5) = -K_G/(Z_G/4);

% Row 5 (soil layer 3)
var_mat(5,4) = -K_G/(Z_G/4);
var_mat(5,5) = ((2*K_G)/(Z_G/4)) ...
              + ((K_G/ALPHA_2)*(Z_G/4));
var_mat(5,6) = -K_G/(Z_G/4);

% Row 6 (soil layer 4)
var_mat(6,5) = -K_G/(Z_G/4);
var_mat(6,6) = ((3 * K_G)/(Z_G/4)) ...
              + ((K_G/ALPHA_2) * (Z_G/4));
var_mat(6,7) = -K_G/(Z_G/8);

% Row 7 (ground)
var_mat(7,6) = -K_G/(Z_G/8);
var_mat(7,7) = soil_Tg_term(avgTgTc, avgTgTsky);
var_mat(7,8) = soil_Tc_term(avgTgTc);

% Row 8 (canopy)
var_mat(8,7) = canopy_Tg_term(avgTgTc);
var_mat(8,8) = canopy_Tc_term(avgTgTc, avgTcTsky);

% Placement of values in solution matrix
sol_mat = zeros(8,1);

% Row 1 (inside)
sol_mat(1,1) = T_inside * h_inside;

% Row 2 (ground)
sol_mat(2,1) = dens_supp * cp_supp * z_supp * T1t1;
```

```
            % Row 3 (soil layer 1)
            sol_mat(3,1) = ((K_G/ALPHA_2) * (Z_G/4)) * T2t1;

            % Row 4 (soil layer 2)
            sol_mat(4,1) = ((K_G/ALPHA_2) * (Z_G/4)) * T3t1;

            % Row 5 (soil layer 3)
            sol_mat(5,1) = ((K_G/ALPHA_2) * (Z_G/4)) * T4t1;

            % Row 6 (soil layer 4)
            sol_mat(6,1) = ((K_G/ALPHA_2) * (Z_G/4)) * T5t1;

            % Row 7 (ground)
            data = [irr Tg rH vap_cond avgTgTsky T_sky t_air];
            sol_mat(7,1) = soil_matrix_ans(data);

            % Row 8 (canopy)
            data = [irr Tc rH vap_cond avgTcTsky T_sky t_air];
            sol_mat(8,1) = canopy_matrix_ans(data);

            % Temperature Array Creation
            temp_mat = zeros(8,1);
            temp_mat = ((var_mat)^-1) * sol_mat;

            % Temperature extraction for linearization
            Tg = temp_mat(7);
            Tc = temp_mat(8);

            % Calculation of average for linearization
            avgTgTc = ((Tg + Tc)/2)^3;
            avgTgTsky = ((Tg + T_sky)/2)^3;
            avgTcTsky = ((Tc + T_sky)/2)^3;

        end % for loop for linearization

        % setting previous temp values in solution matrix
        T0t1 = temp_mat(1);
        T1t1 = temp_mat(2);
        T2t1 = temp_mat(3);
        T3t1 = temp_mat(4);
        T4t1 = temp_mat(5);
        T5t1 = temp_mat(6);
        Tgt1 = temp_mat(7);
        Tct1 = temp_mat(8);

        % Exporting temperature array
        fprintf(fid,...
 '%6.2f\t %6.2f\t %6.2f\t %6.2f\t %6.2f\t %6.2f\t %6.2f\t %6.2f\n', ...
                T0t1, T1t1, T2t1, T3t1, T4t1, T5t1, Tgt1, Tct1);

    end % for loop for hours

%          fprintf(fid, '\n');
```

```
    end % for loop for each trial day

    status = fclose(fid);

% end GreenRoofModel.m
```

```
function soilTgTerm = soil_Tg_term(avgTgTc, avgTgTsky)
% SOIL_TG_TERM is a function to receive environmental parameters and
% return a value for the coefficient of ground temperature for the soil
% layer matrix equation.
%
% MEng Project: A model to optimize performance of green roof
%               structures.  Copyright 2006 William Striar Lambert
% Author: William Lambert
% Original code: October 23, 2005
% Modifications: December 7, 2005
%                December 14, 2005
%                December 20, 2005
%                January 25, 2006
%                January 28, 2006
%                February 6, 2006
%
% calling sequence:
%   soilTgTerm = soil_Tg_term(avgTgTc, avgTgTsky)
%
% =====================
% Variable Dictionary
% =====================
%   A:          cube of the average ground and canopy temperatures
%                   A = (( Tg + Tc ) / 2) ^ 3  (K^3)
%   B:          cube of the average ground and sky temperatures
%                   B = (( Tg + Tsky ) / 2) ^ 3 (K^3)
%   SIGMA:      Stefan-Boltzmann constant (J m^-2 hr^-1 K^-4) (global)
%   EPSILON_G:  emissivity of ground (global)
%   e_kLAI:     exp(-ext_coeff * LAI)  (global)
%   H_AIR:      convective heat transfer coefficient to air
%               (J m^-2 hr^-1 K^-1) (global)
%   avgTgTc:    A value sent from calling sequence (K^3)
%   avgTgTsky:  B value sent from calling sequence (K^3)
%
%   VARIED FOR SENSITIVITY ANALYSIS
%   --------------------------------
%   LAI:        leaf area index (global)
%   Z_G:        soil depth (m) (global)
%   ALPHA_2:    thermal diffusivity of ground (m^2/hr)  (global)
%               (From "Modeling plant and soil systems" p. 405)

    global RHO_G e_kLAI Z_G K_G ALPHA_2 LAT_VAP PRESSURE ...
        RHO_C EPSILON_C SIGMA EPSILON_G ABS_G ...
        H_AIR LAI

% Equation
%     if (LAI < 1)
%        soilTgTerm = (4 * EPSILON_G * SIGMA * (((1 - LAI) * B) ...
```

```
%                       + (LAI * A))) + H_AIR ...
%                       + (K_G/(Z_G/8));
%     else
%        soilTgTerm = (4 * EPSILON_G * SIGMA * A) + H_AIR ...
%                       + (K_G/(Z_G/8));
%     end %if

% Parameter Values obtained from input data vector
    A = avgTgTc;
    B = avgTgTsky;

% Calculate coefficient of ground temperature for soil layer
    if (LAI < 1)
       soilTgTerm = (4 * EPSILON_G * SIGMA * (((1 - LAI) * B) ...
                      + (LAI * A))) + H_AIR ...
                      + (K_G/(Z_G/8));
    else
       soilTgTerm = (4 * EPSILON_G * SIGMA * A) + H_AIR ...
                      + (K_G/(Z_G/8));
    end %if

end % function soil_Tg_term
```

```
function soilTcTerm = soil_Tc_term(temp)
% SOIL_TC_TERM is a function to receive environmental parameters and
% return a value for the coefficient of canopy temperature for the soil
% layer matrix equation.
%
% MEng Project: A model to optimize performance of green roof
%               structures.  Copyright 2006 William Striar Lambert
% Author: William Lambert
% Original code: October 23, 2005
% Modifications: December 7, 2005
%                December 14, 2005
%                January 25, 2006
%                January 28, 2006
%                February 6, 2006
%
% calling sequence:
%   soilTcTerm = soil_Tc_term(temp)
%       where temp is the value of A defined below
%
% =====================
% Variable Dictionary
% =====================
%   A:          cube of the average ground and canopy temperatures
%                   A = (( Tg + Tc ) / 2) ^ 3  (K^3)
%   SIGMA:      Stefan-Boltzmann constant (J m^-2 hr^-1 K^-4) (global)
%   EPSILON_G:  emissivity of ground (global)
%   e_kLAI:     exp(-ext_coeff * LAI)  (global)
%   temp:       A value sent from calling sequence (K^3)

%   VARIED FOR SENSITIVITY ANALYSIS
%   -------------------------------
%   LAI:        leaf area index (global)
```

```
    global RHO_G e_kLAI Z_G K_G ALPHA_2 LAT_VAP PRESSURE ...
        RHO_C EPSILON_C SIGMA EPSILON_G ABS_G ...
        H_AIR LAI

% Equation
%     if (LAI < 1)
%         soilTcTerm = -4 * LAI * EPSILON_G * SIGMA * A;
%     else
%         soilTcTerm = -4 * EPSILON_G * SIGMA * A;
%     end %if

% Parameter Values obtained from input data
    A = temp;

% Calculate coefficient of canopy temperature for soil layer
    if (LAI < 1)
        soilTcTerm = -4 * LAI * EPSILON_G * SIGMA * A;
    else
        soilTcTerm = -4 * EPSILON_G * SIGMA * A;
    end %if

end % function soil_Tc_term
```

```
function canopyTgTerm = canopy_Tg_term(temp)
% CANOPY_TG_TERM is a function to receive environmental parameters and
% return a value for the coefficient of ground temperature for the
% canopy layer matrix equation.
%
% MEng Project: A model to optimize performance of green roof
% structures.  Copyright 2006 William Striar Lambert
% Author: William Lambert
% Original code: November 7, 2005
% Modifications: December 7, 2005
%                December 14, 2005
%                January 25, 2006
%                January 28, 2006
%                February 6, 2006
%
% calling sequence:
%   canopyTgTerm = canopy_Tg_term(temp)
%       where temp is the value of A defined below
%
% =====================
% Variable Dictionary
% =====================
%   A:          cube of the average canopy and ground temperatures
%                   A = (( Tc + Tg ) / 2) ^ 3   (K^3)
%   EPSILON_C:  emissivity of canopy (global)
%   EPSILON_G:  emissivity of ground (global)
%   SIGMA:      Stefan-Boltzmann constant (J m^-2 hr^-1 K^-4) (global)
%   e_kLAI:     exp(-ext_coeff * LAI)  (global)
%   temp:       A value sent from calling sequence  (K^3)
%
%   VARIED FOR SENSITIVITY ANALYSIS
```

```
%   --------------------------------
%   LAI:         leaf area index (global)

    global RHO_G e_kLAI Z_G K_G ALPHA_2 LAT_VAP PRESSURE ...
        RHO_C EPSILON_C SIGMA EPSILON_G ABS_G ...
        H_AIR LAI

% Equation
%     if (LAI < 1)
%         canopyTgTerm = -4 * SIGMA * A * ((LAI * EPSILON_G) ...
%                          + EPSILON_C);
%     else
%         canopyTgTerm = -4 * SIGMA * A * (EPSILON_G + EPSILON_C);
%     end %if

% Parameter Values obtained from input data
    A = temp;

% Calculate coefficient of ground temperature for canopy layer
    if (LAI < 1)
        canopyTgTerm = -4 * SIGMA * A * ((LAI * EPSILON_G) ...
                         + EPSILON_C);
    else
        canopyTgTerm = -4 * SIGMA * A * (EPSILON_G + EPSILON_C);
    end %if

end % function canopy_Tg_term
```

```
function canopyTcTerm = canopy_Tc_term(temp1, temp2)
% CANOPY_TC_TERM is a function to receive environmental parameters and
% return a value for the coefficient of canopy temperature for the
% canopy layer matrix equation.
%
% MEng Project: A model to optimize performance of green roof
% structures.  Copyright 2006 William Striar Lambert
% Author: William Lambert
% Original code: November 7, 2005
% Modifications: December 7, 2005
%                December 14, 2005
%                December 20, 2005
%                December 21, 2005
%                January 25, 2006
%                January 28, 2006
%                February 6, 2006
%
% calling sequence:
%   canopyTcTerm = canopy_Tc_term(enviro_parameter_vector)
%   where parameter_vector holds two data values as defined below:
%       (A, C)
%
% =====================
% Variable Dictionary
% =====================
%   C:           cube of the average canopy and sky temperatures
%                    C = (( Tc + Tsky ) / 2) ^ 3    (K^3)
```

```
%   A:          cube of the average canopy and ground temperatures
%                   A = (( Tc + Tg ) / 2) ^ 3    (K^3)
%   EPSILON_C:  emissivity of canopy (global)
%   EPSILON_G:  emissivity of ground (global)
%   SIGMA:      Stefan-Boltzmann constant (J m^-2 hr^-1 K^-4) (global)
%   H_AIR:      convective heat transfer coefficient to air
%               (J m^-2 hr^-1 K^-1) (global)
%   temp1:       A value sent from calling sequence  (K^3)
%   temp2:       C value sent from calling sequence  (K^3)
%
%   VARIED FOR SENSITIVITY ANALYSIS
%   --------------------------------
%   LAI:        leaf area index (global)

    global RHO_G e_kLAI Z_G K_G ALPHA_2 LAT_VAP PRESSURE ...
        RHO_C EPSILON_C SIGMA EPSILON_G ABS_G ...
        H_AIR LAI

% Equation
%     if (LAI < 1)
%         canopyTcTerm = 4 * SIGMA * ((LAI * EPSILON_G * A) ...
%                             + (EPSILON_C * (C + A))) + H_AIR;
%     else
%         canopyTcTerm = 4 * SIGMA * ((EPSILON_G * A) ...
%                             + (EPSILON_C * (C + A))) + H_AIR;
%     end %if

% Parameter Values obtained from input vector
    A = temp1;
    C = temp2;

% Calculate coefficient of canopy temperature for canopy layer
    if (LAI < 1)
        canopyTcTerm = 4 * SIGMA * ((LAI * EPSILON_G * A) ...
                            + (EPSILON_C * (C + A))) + H_AIR;
    else
        canopyTcTerm = 4 * SIGMA * ((EPSILON_G * A) ...
                            + (EPSILON_C * (C + A))) + H_AIR;
    end %if

end % function canopy_Tc_term
```

```
function ansMatrixSoil = soil_matrix_ans(all_data)
% SOIL_MATRIX_ANS is a function to receive environmental parameters and
% return the answer to the right matrix for the soil heat balance
% equation.
%
% MEng Project: A model to optimize performance of green roof
% structures.  Copyright 2006 William Striar Lambert
% Author: William Lambert
% Original code: November 13, 2005
% Modifications: December 7, 2005
%                December 9, 2005
%                December 14, 2005
%                January 26, 2006
```

```
%                 January 28, 2006
%                 February 6, 2006
%
% calling sequence:
%   ansMatrixSoil = soil_matrix_ans(enviro_parameter_vector)
%   where parameter_vector holds seven data values as defined below:
%       (irr, T_g, rH, g_v, B, T_sky, T_air)
%
% =====================
% Variable Dictionary
% =====================
%   all_data:       vector of input data
%   irr:            global horizontal irradiance (J/m^2)
%   T_g:            ground temperature  (K)
%   rH:             relative humidity
%   sat_vap_press:  saturation vapor pressure at ground temperature
%                   (Pa)
%   vap_press:      vapor pressure, rH * sat_vap_press  (Pa)
%   g_v:        conductance for vapor (mol m^-2 hr^-1):
%               (From Campbell 234)
%               Equation:  (0.12 * u)/(0.6 + 0.2 * u);
%               u = wind speed (m/s)
%   LAT_VAP:    latent heat of vaporization of water (J/mol)
%               (From CB 37) (global)
%   PRESSURE:   atmospheric pressure at STP (Pa) (global)
%   RHO_C:      reflectivity of canopy (grass) (From CB 172) (global)
%   ABS_G:      absorbtivity of ground:  1 - RHO_G  (global)
%   e_kLAI:     exp(-ext_coeff * LAI)  (global)
%   a1...a7:    coefficients in the saturation vapor pressure equation
%   B:          cube of the average ground and sky temperatures
%                   B = (( Tg + Tsky ) / 2) ^ 3   (K^3)
%   T_sky:      sky temperature  (K)
%   T_air:      air temperature  (K)
%   H_AIR:      convective heat transfer coefficient to air
%               (J m^-2 hr^-1 K^-1)    (global)
%   e_kLAI:     exp(-ext_coeff * LAI)  (global)
%
%   VARIED FOR SENSITIVITY ANALYSIS
%   -------------------------------
%   LAI:        leaf area index (global)

    global RHO_G e_kLAI Z_G K_G ALPHA_2 LAT_VAP PRESSURE ...
        RHO_C EPSILON_C SIGMA EPSILON_G ABS_G ...
        H_AIR LAI

% Equation
%     if (LAI < 1)
%         ansMatrixSoil = ((1 - RHO_C) * irr * e_kLAI * ABS_G) ...
%                          + (H_AIR * T_air) + (4 * (1 - LAI) ...
%                          * EPSILON_G * SIGMA * B * T_sky) ...
%                          - ((LAT_VAP * g_v * sat_vap_press)...
%                          /PRESSURE) ...
%                          + ((LAT_VAP * g_v * vap_press) ...
%                          /PRESSURE);
%     else
%         ansMatrixSoil = ((1 - RHO_C) * irr * e_kLAI * ABS_G) ...
```

```
%                            + (H_AIR * T_air) - ((LAT_VAP * g_v ...
%                            * sat_vap_press)/PRESSURE) ...
%                            + ((LAT_VAP * g_v * vap_press)/PRESSURE);
%     end %if

% Parameters
    a1 = -5.8002206e3;
    a2 = 1.3914993;
    a3 = -48.640238e-3;
    a4 = 41.764768e-6;
    a5 = -14.452093e-9;
    a6 = 0.0;
    a7 = 6.5459673;

% Parameter Values obtained from input data vector
    irr = all_data(1);
    T_g = all_data(2);
    rH = all_data(3);
    g_v = all_data(4);
    B = all_data(5);
    T_sky = all_data(6);
    T_air = all_data(7);

%  NEED TO CALCULATE SATURATION VAPOR PRESSURE AND VAPOR PRESSURE

    sat_vap_press = exp(a1/T_g + a2 + a3 * T_g + a4 * T_g^2 + ...
                        a5 * T_g^3 + a6 * T_g^4 + a7 * log(T_g));

    vap_press = sat_vap_press * rH;

% Calculate answer for right matrix for soil layer
    if (LAI < 1)
        ansMatrixSoil = ((1 - RHO_C) * irr * e_kLAI * ABS_G) ...
                        + (H_AIR * T_air) + (4 * (1 - LAI) ...
                        * EPSILON_G * SIGMA * B * T_sky) ...
                        - ((LAT_VAP * g_v * sat_vap_press)...
                        /PRESSURE) ...
                        + ((LAT_VAP * g_v * vap_press) ...
                        /PRESSURE);
    else
        ansMatrixSoil = ((1 - RHO_C) * irr * e_kLAI * ABS_G) ...
                        + (H_AIR * T_air) - ((LAT_VAP * g_v ...
                        * sat_vap_press)/PRESSURE) ...
                        + ((LAT_VAP * g_v * vap_press)/PRESSURE);
    end %if

end % function soil_matrix_ans
```

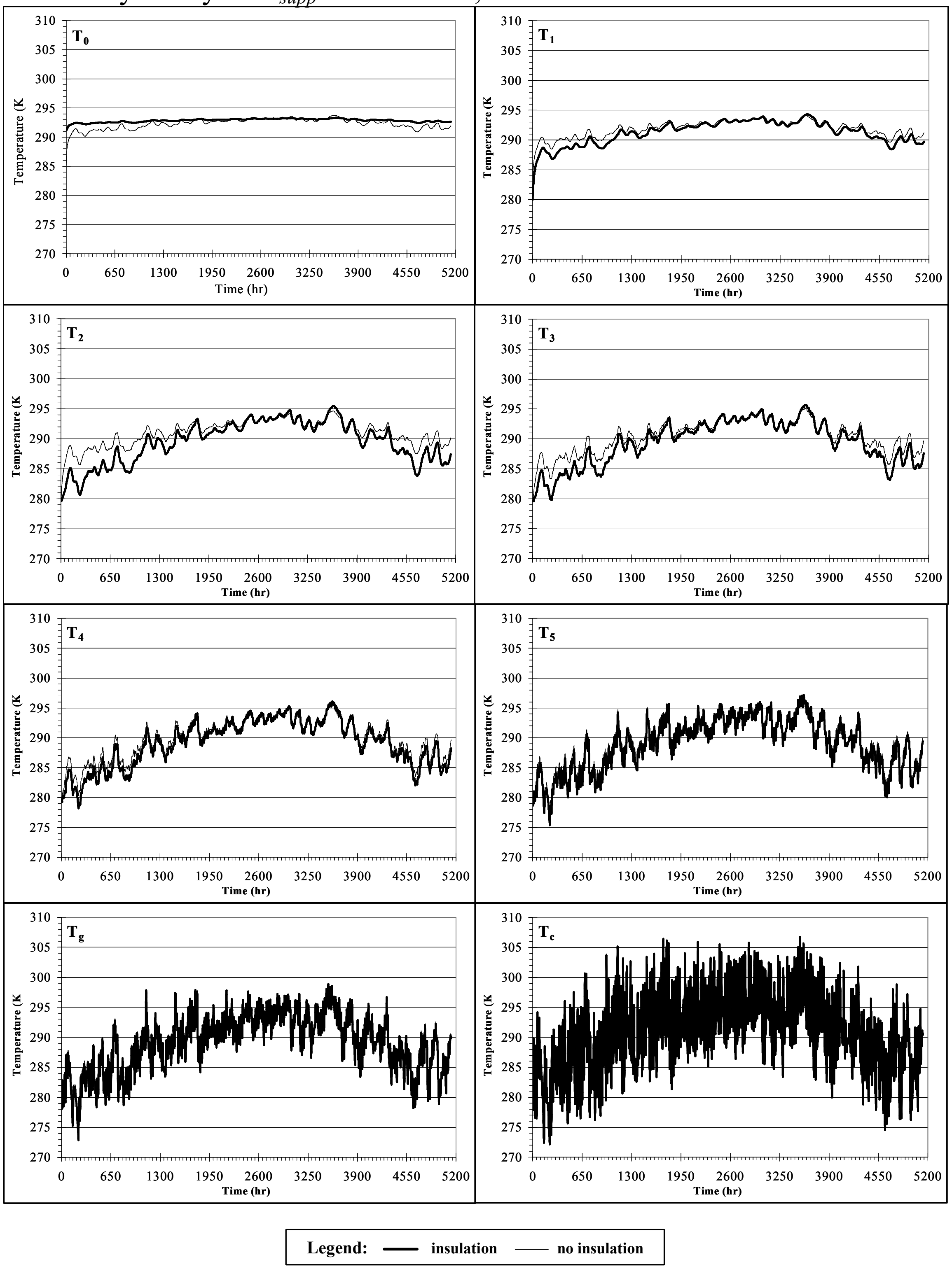
```
function ansMatrixCanopy = canopy_matrix_ans(all_data)
% CANOPY_MATRIX_ANS is a function to receive environmental parameters
% and return the answer to the right matrix for the canopy heat balance
% equation.
%
% MEng Project: A model to optimize performance of green roof
% structures.  Copyright 2006 William Striar Lambert
```

```
% Author: William Lambert
% Original code: November 13, 2005
% Modifications: December 7, 2005
%                December 9, 2005
%                December 14, 2005
%                January 26, 2006
%                February 6, 2006
%
% calling sequence:
%   ansMatrixCanopy = canopy_matrix_ans(enviro_parameter_vector)
%   where parameter_vector holds seven data values as defined below:
%       (irr, T_c, rH, g_v, C, T_sky, T_air)
%
% =====================
% Variable Dictionary
% =====================
%   all_data:       vector of input data
%   irr:            global horizontal irradiance (J/m^2)
%   T_c:            canopy temperature  (K)
%   rH:             relative humidity
%   sat_vap_press:  saturation vapor pressure at canopy temperature
%                   (Pa)
%   vap_press:      vapor pressure, rH * sat_vap_press   (Pa)
%   g_v:        conductance for vapor (mol m^-2 hr^-1):
%               (From Campbell 234)
%               Equation:  (0.12 * u)/(0.6 + 0.2 * u);
%               u = wind speed (m/s)
%   LAT_VAP:    latent heat of vaporization of water (J/mol)
%               (From CB 37)  (global)
%   PRESSURE:   atmospheric pressure at STP (Pa) (global)
%   RHO_C:      reflectivity of canopy (grass) (From CB 172) (global)
%   SIGMA:      Stefan-Boltzmann constant (J m^-2 hr^-1 K^-4) (global)
%   e_kLAI:     exp(-ext_coeff * LAI)  (global)
%   a1...a7:    coefficients in the saturation vapor pressure equation
%   C:          cube of the average canopy and sky temperatures
%                   C = (( Tc + Tsky ) / 2) ^ 3  (K^3)
%   T_sky:      sky temperature  (K)
%   T_air:      air temperature  (K)
%   H_AIR:      convective heat transfer coefficient to air
%               (J m^=2 hr^-1 K^-1)  (global)

    global RHO_G e_kLAI Z_G K_G ALPHA_2 LAT_VAP PRESSURE ...
        RHO_C EPSILON_C SIGMA EPSILON_G ABS_G ...
        H_AIR LAI

% Equation
% ansMatrixCanopy = (irr * (1 - RHO_C) * (1 - e_kLAI)) ...
%                         + (H_AIR * T_air)...
%                         + (4 * EPSILON_C * SIGMA * C * T_sky) ...
%                         - ((LAT_VAP * g_v * sat_vap_press) ...
%                         /PRESSURE) ...
%                         + ((LAT_VAP * g_v * vap_press)/PRESSURE;

% Parameters
    a1 = -5.8002206e3;
    a2 = 1.3914993;
```

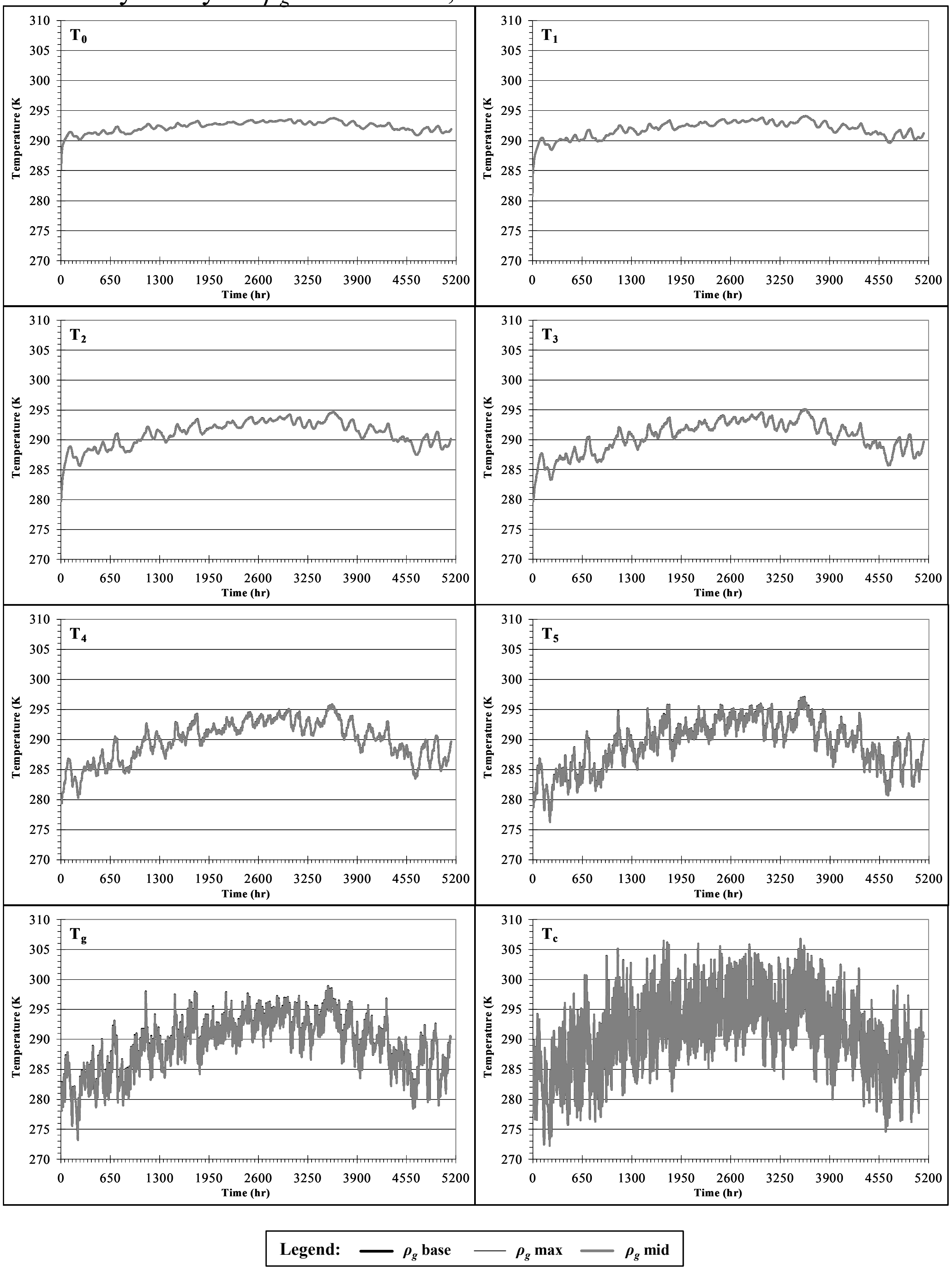
```
    a3 = -48.640238e-3;
    a4 = 41.764768e-6;
    a5 = -14.452093e-9;
    a6 = 0.0;
    a7 = 6.5459673;

% Parameter Values obtained from input data vector
    irr = all_data(1);
    T_c = all_data(2);
    rH = all_data(3);
    g_v = all_data(4);
    C = all_data(5);
    T_sky = all_data(6);
    T_air = all_data(7);

%  NEED TO CALCULATE SATURATION VAPOR PRESSURE AND VAPOR PRESSURE

    sat_vap_press = exp(a1/T_c + a2 + a3 * T_c + a4 * T_c^2 + ...
                        a5 * T_c^3 + a6 * T_c^4 + a7 * log(T_c));

    vap_press = sat_vap_press * rH;

% Calculate answer for right matrix for canopy layer
  ansMatrixCanopy = (irr * (1 - RHO_C) * (1 - e_kLAI)) ...
                        + (H_AIR * T_air)...
                        + (4 * EPSILON_C * SIGMA * C * T_sky) ...
                        -  ((LAT_VAP * g_v * sat_vap_press) ...
                        /PRESSURE) ...
                        + ((LAT_VAP * g_v * vap_press)/PRESSURE);

end % function canopy_matrix_ans
```

Appendix E

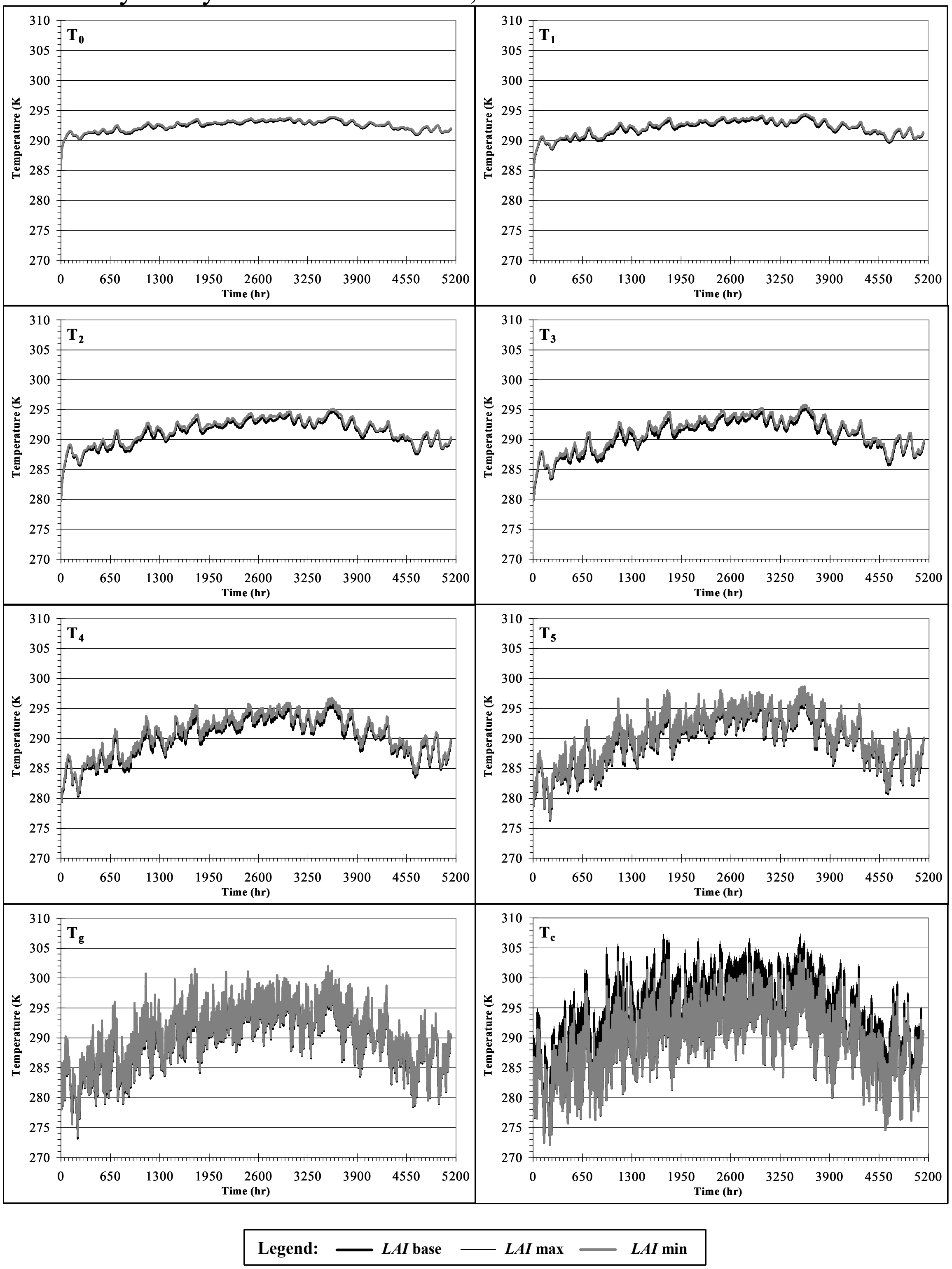
Sensitivity Analysis: k_{supp} – New York, NY



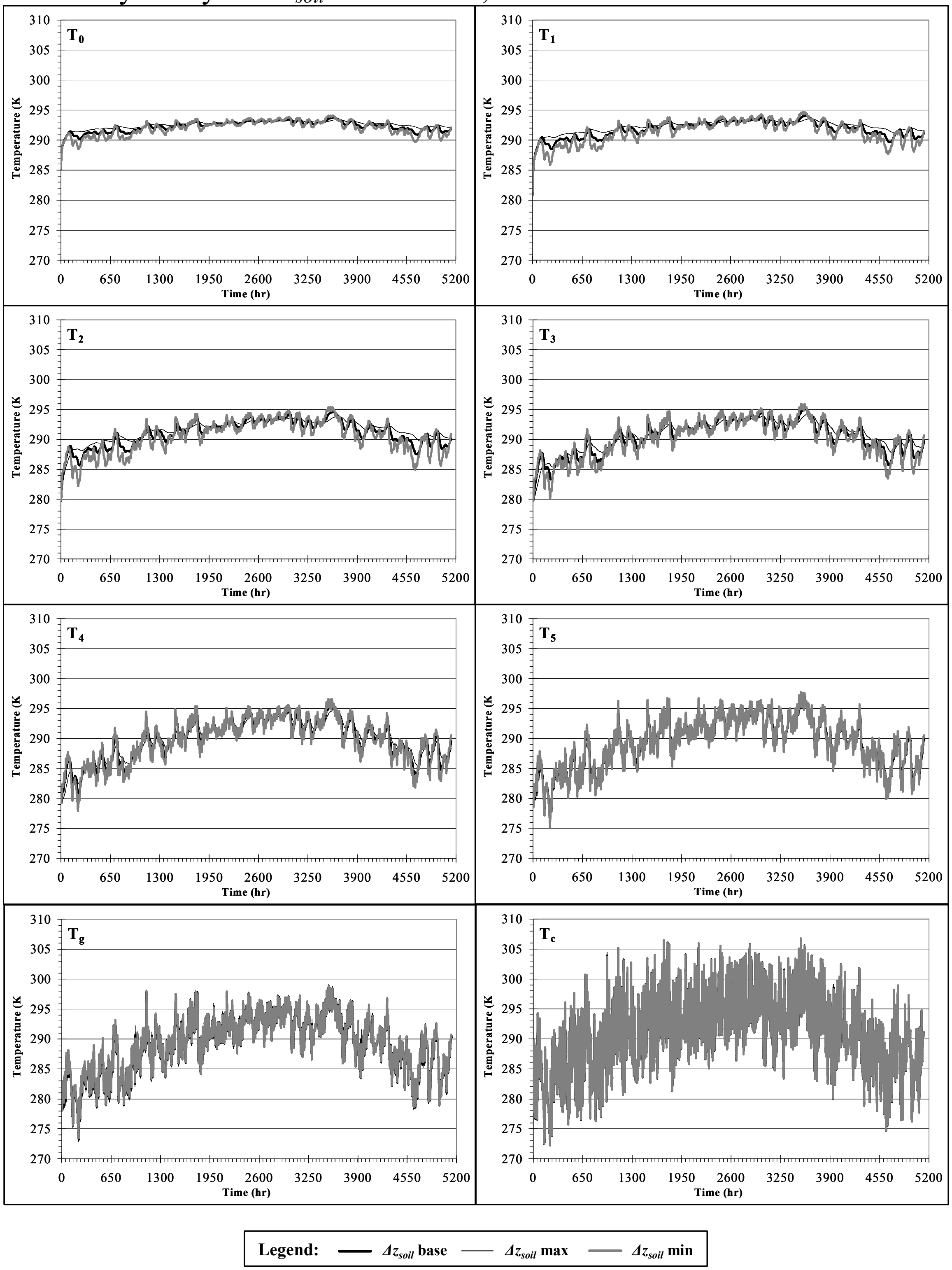
Sensitivity Analysis: ρ_g – New York, NY



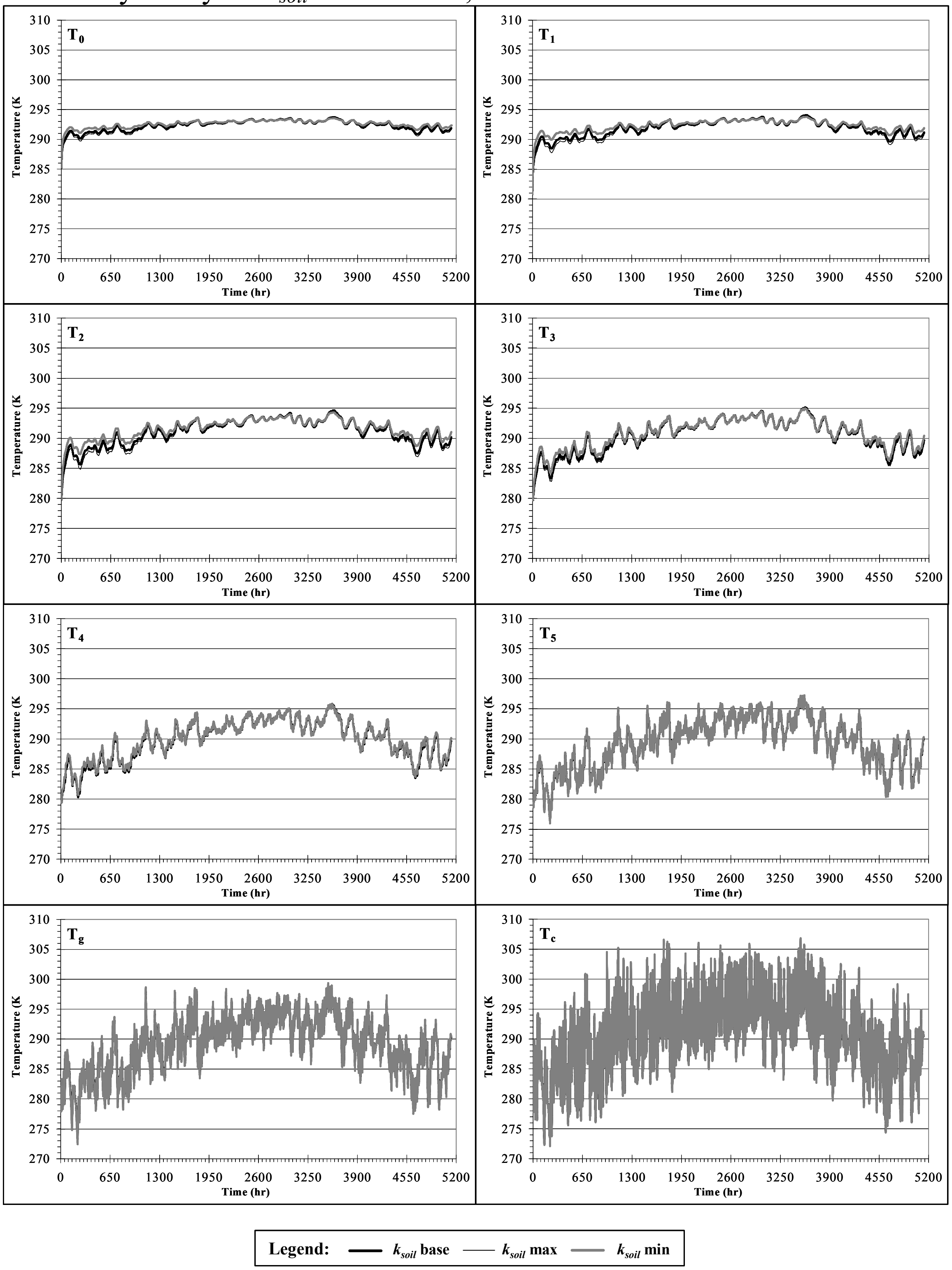
Sensitivity Analysis: *LAI* – New York, NY



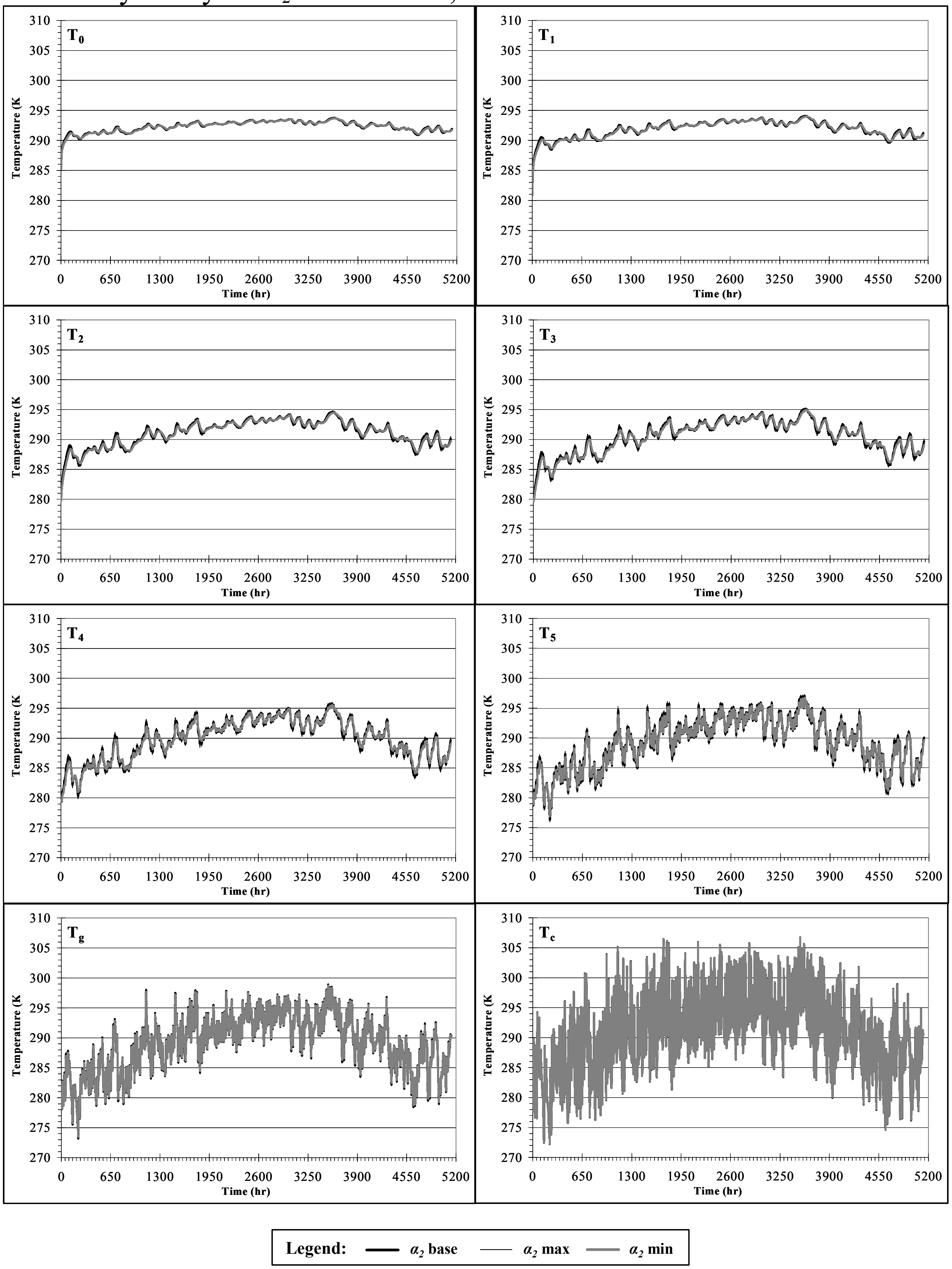
Sensitivity Analysis: Δz_{soil} – New York, NY



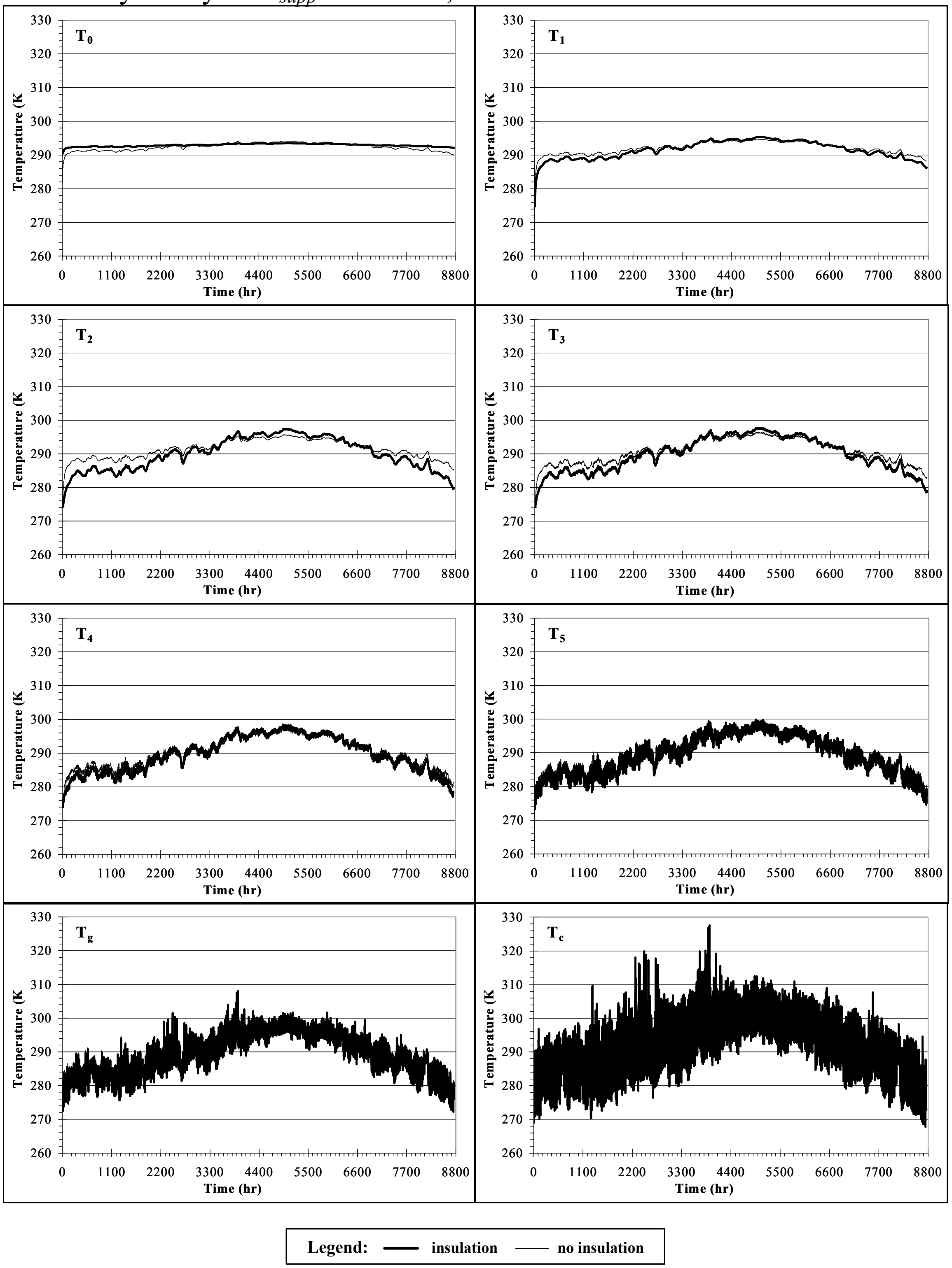
Sensitivity Analysis: k_{soil} – New York, NY



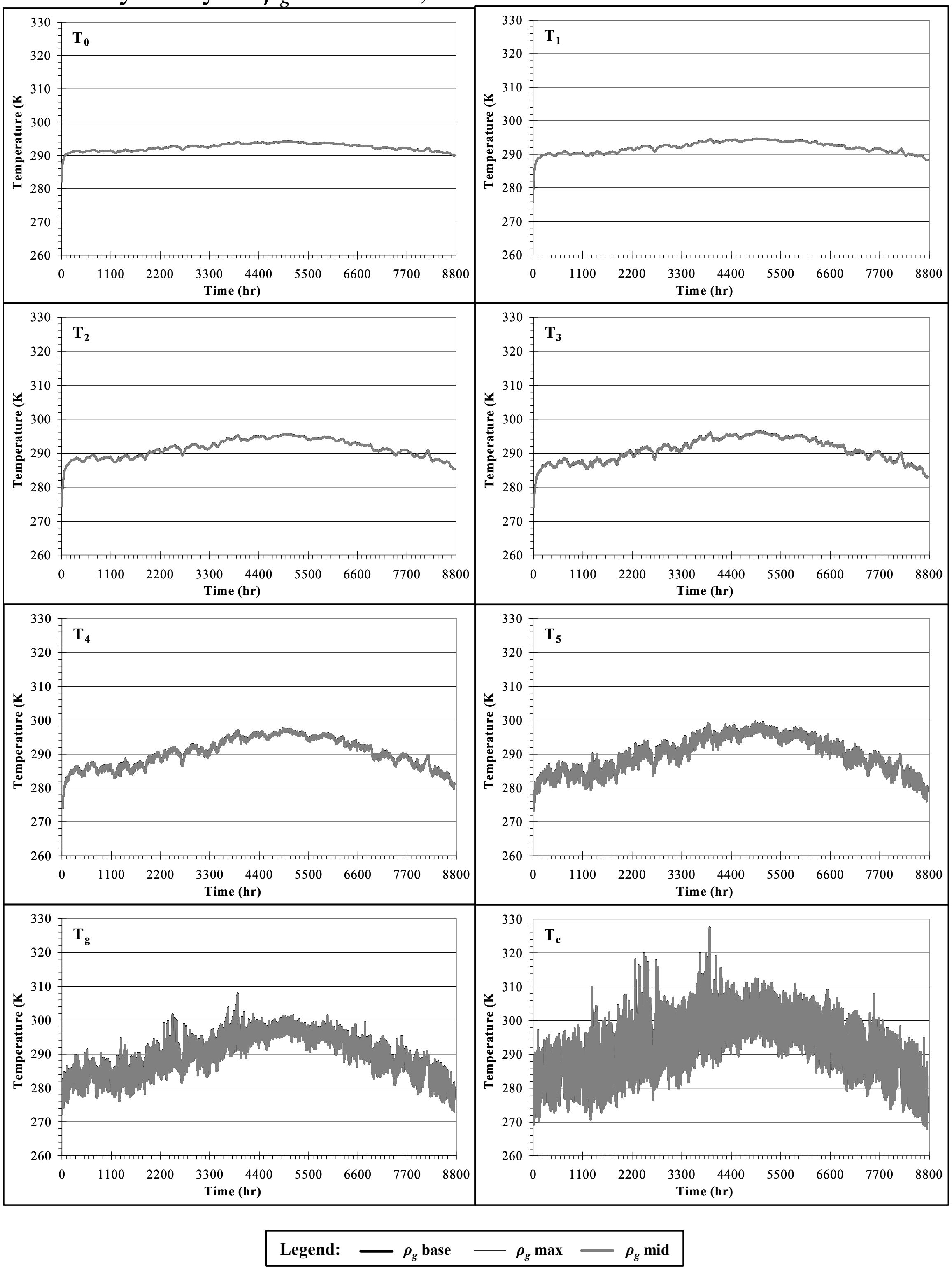
Sensitivity Analysis: α_2 – New York, NY



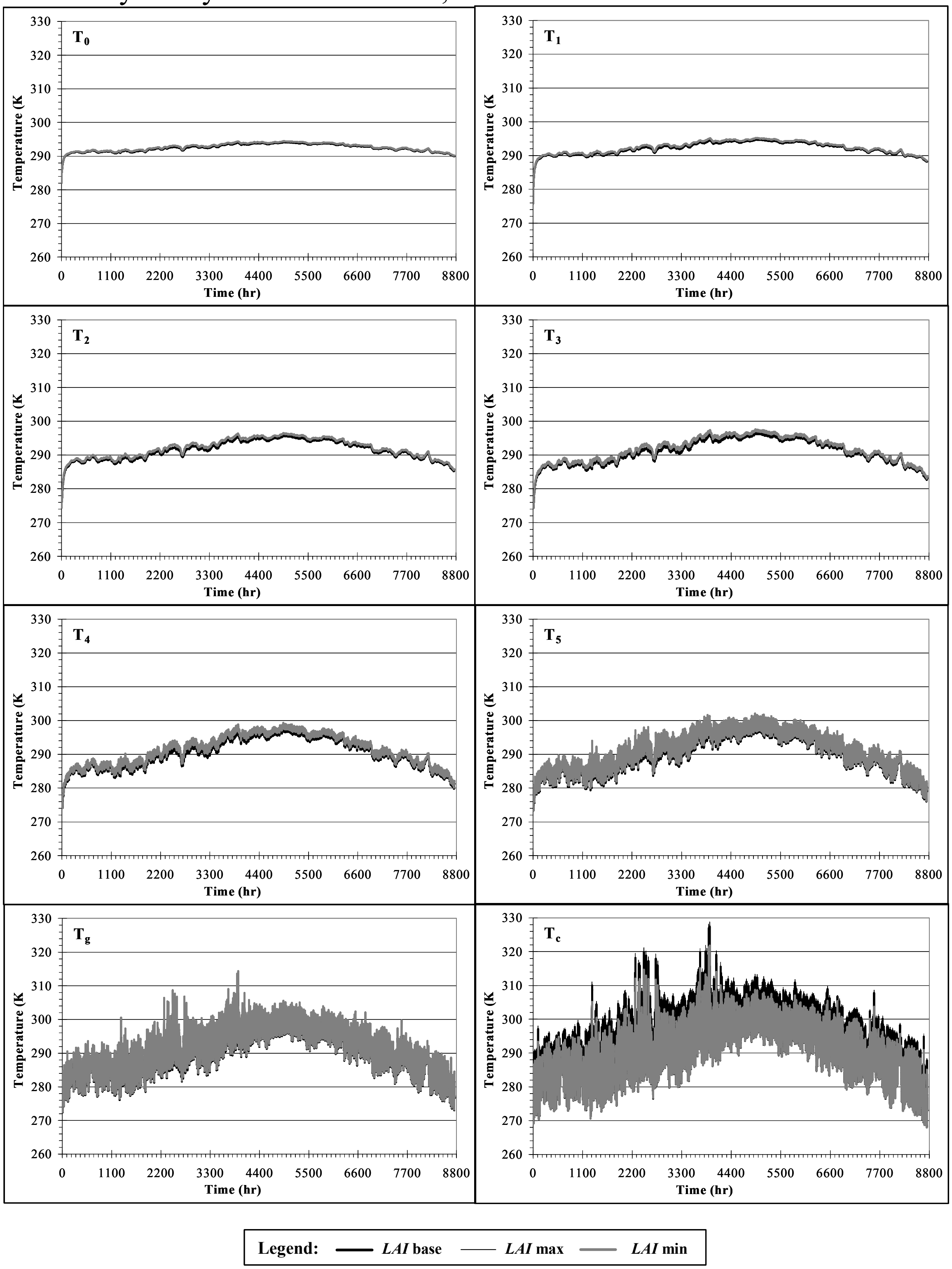
Sensitivity Analysis: k_{supp} – Phoenix, AZ



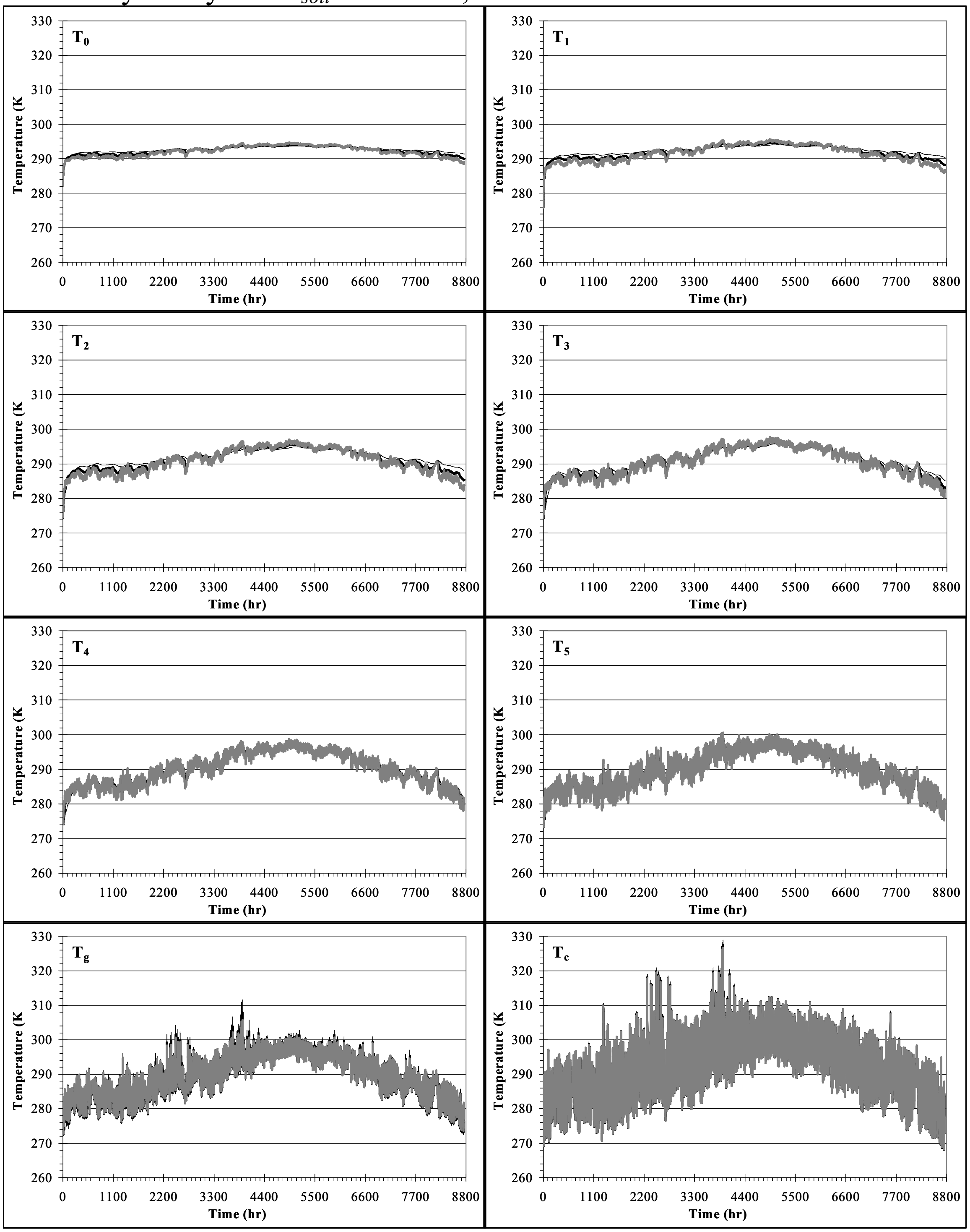
Sensitivity Analysis: ρ_g – Phoenix, AZ



Sensitivity Analysis: *LAI* – Phoenix, AZ

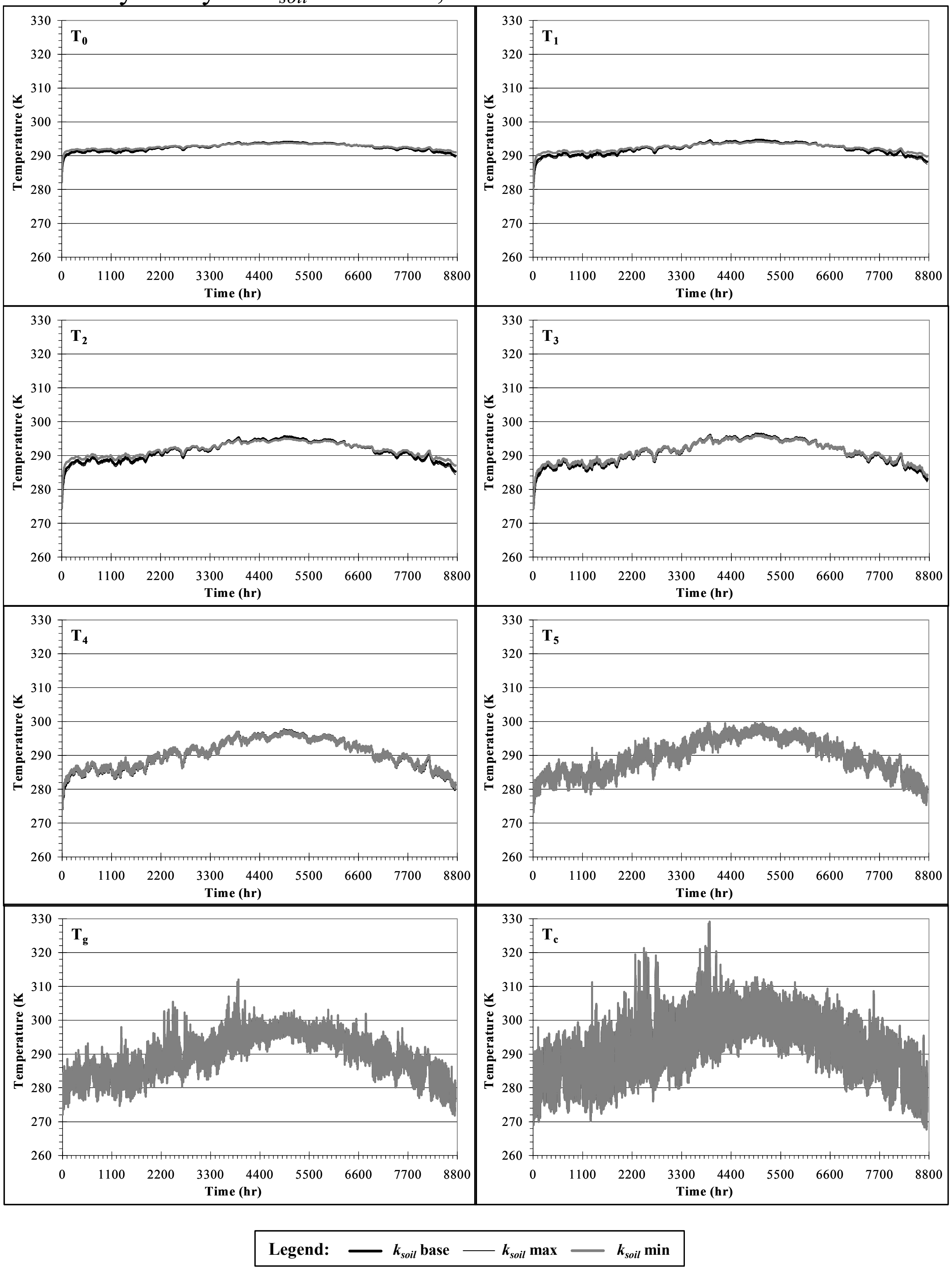


Sensitivity Analysis: Δz_{soil} – Phoenix, AZ

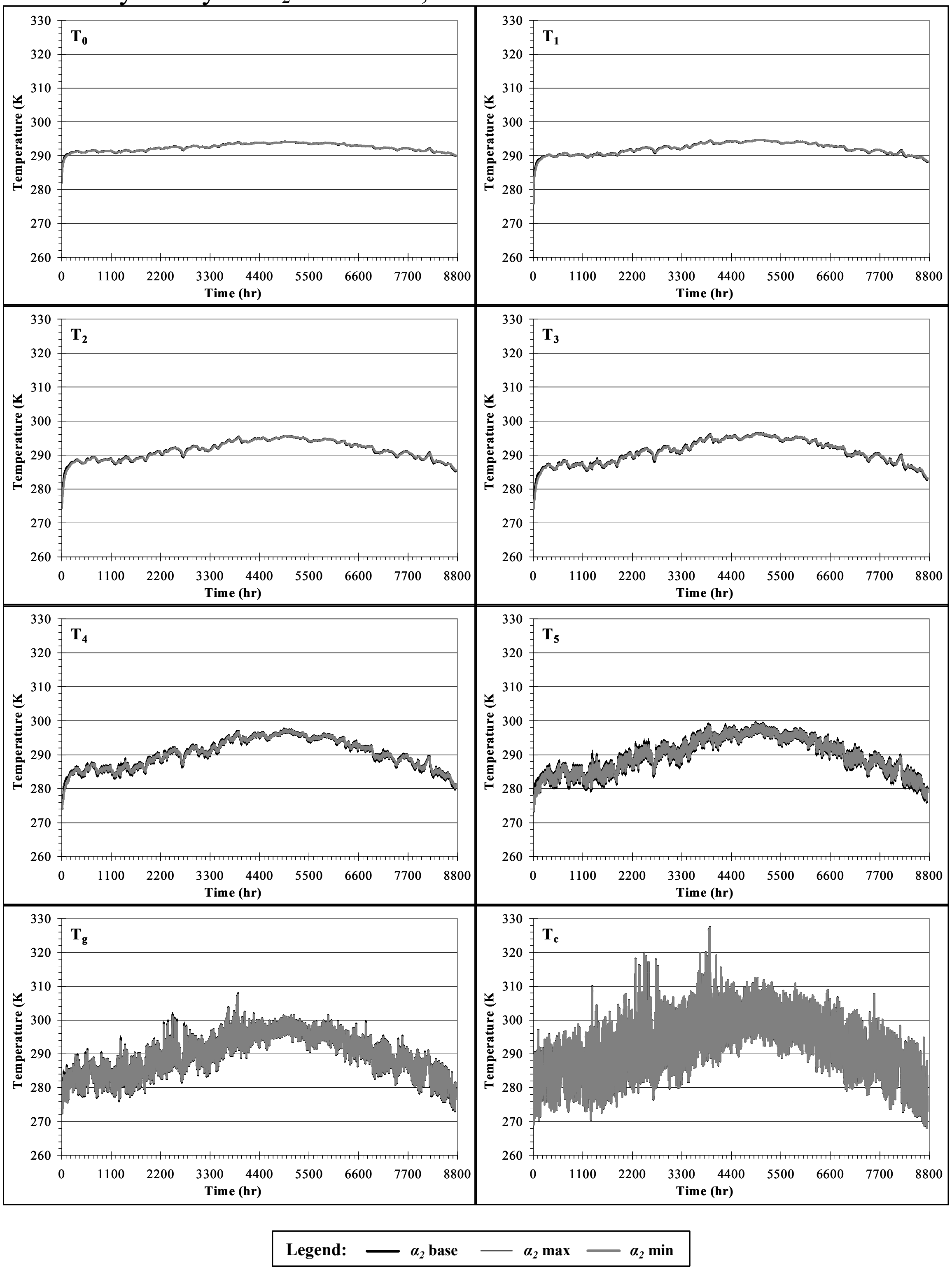


Legend: Δz_{soil} base Δz_{soil} max Δz_{soil} min

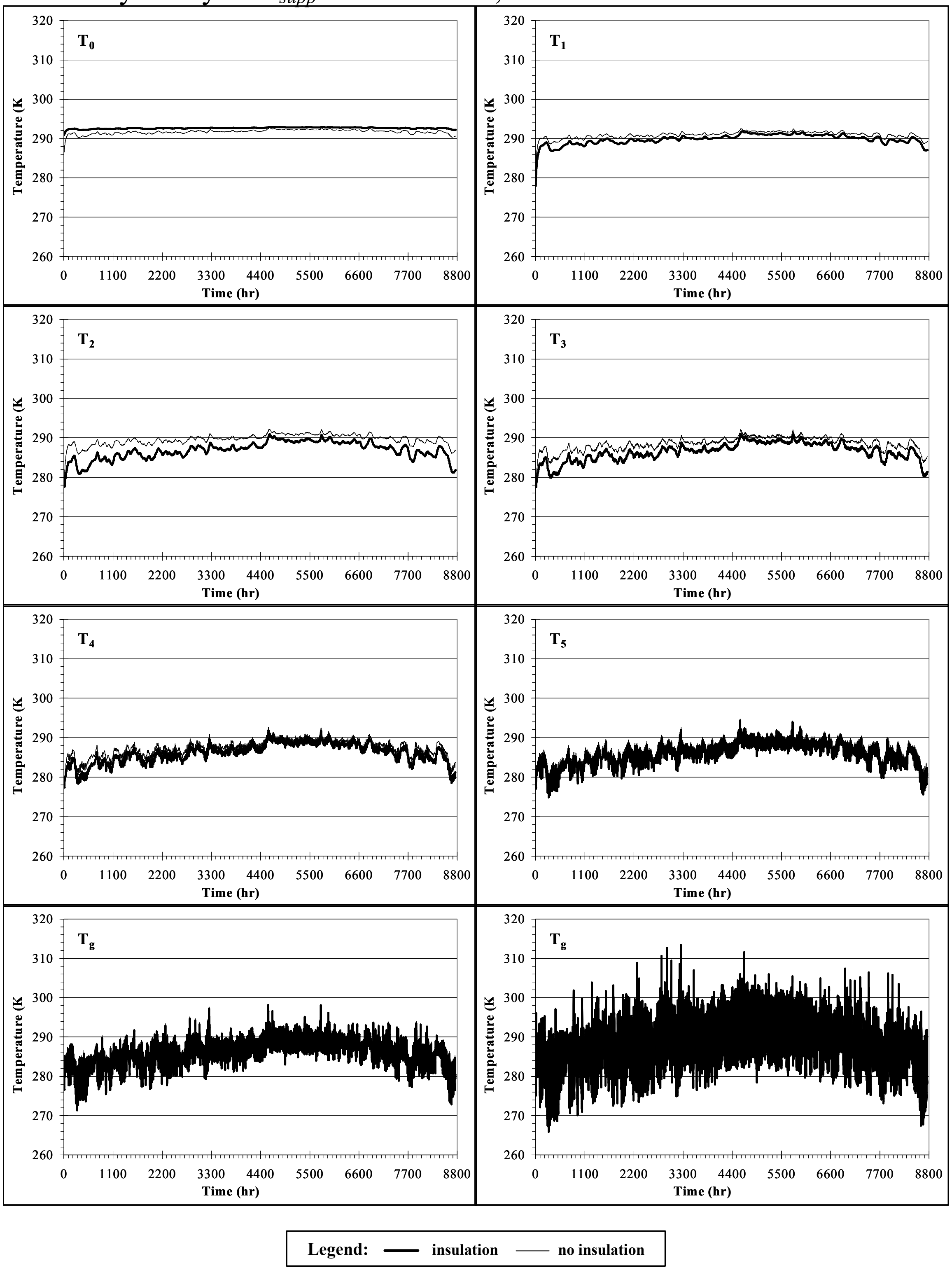
Sensitivity Analysis: k_{soil} – Phoenix, AZ



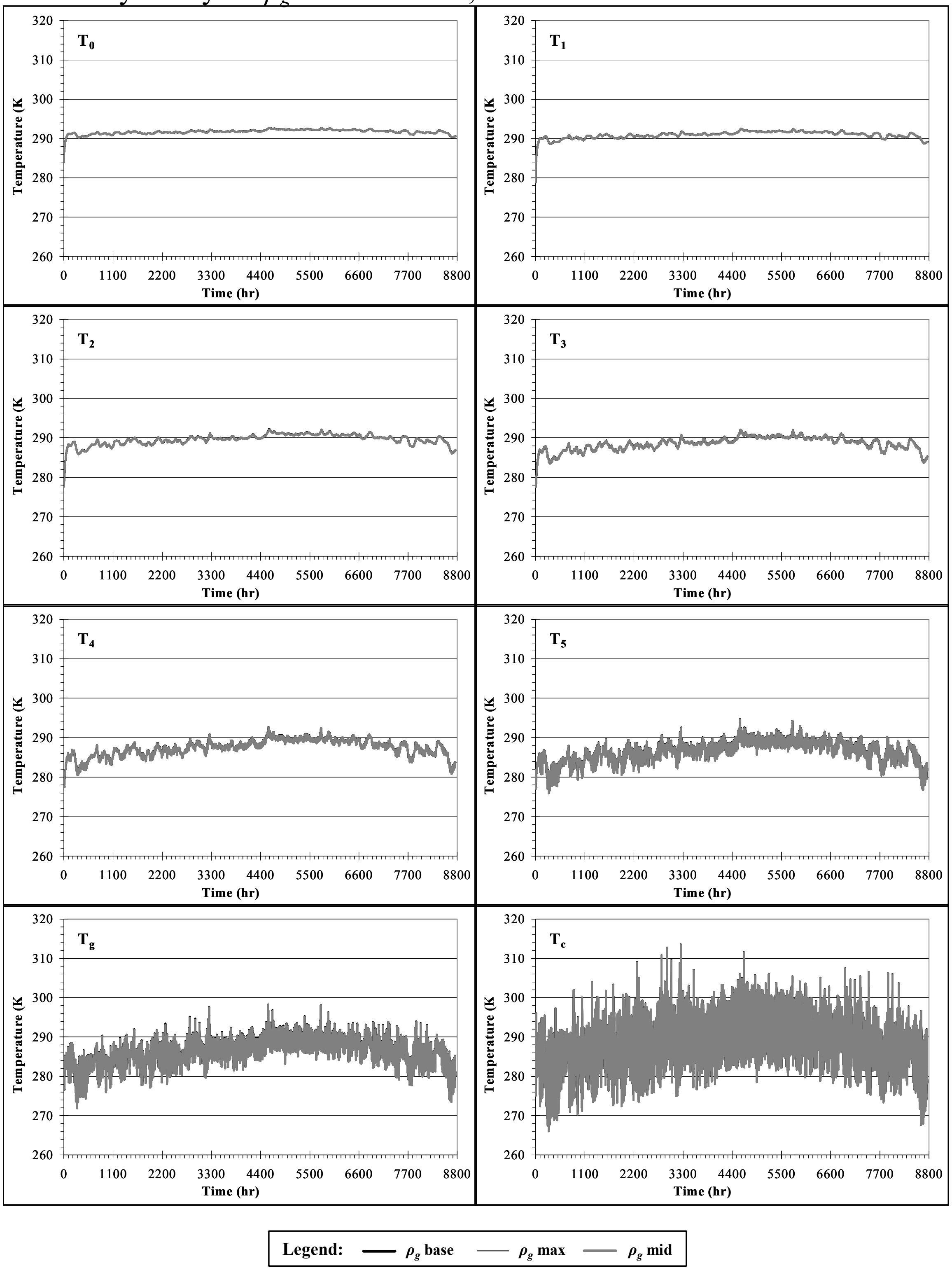
Sensitivity Analysis: α_2 – Phoenix, AZ



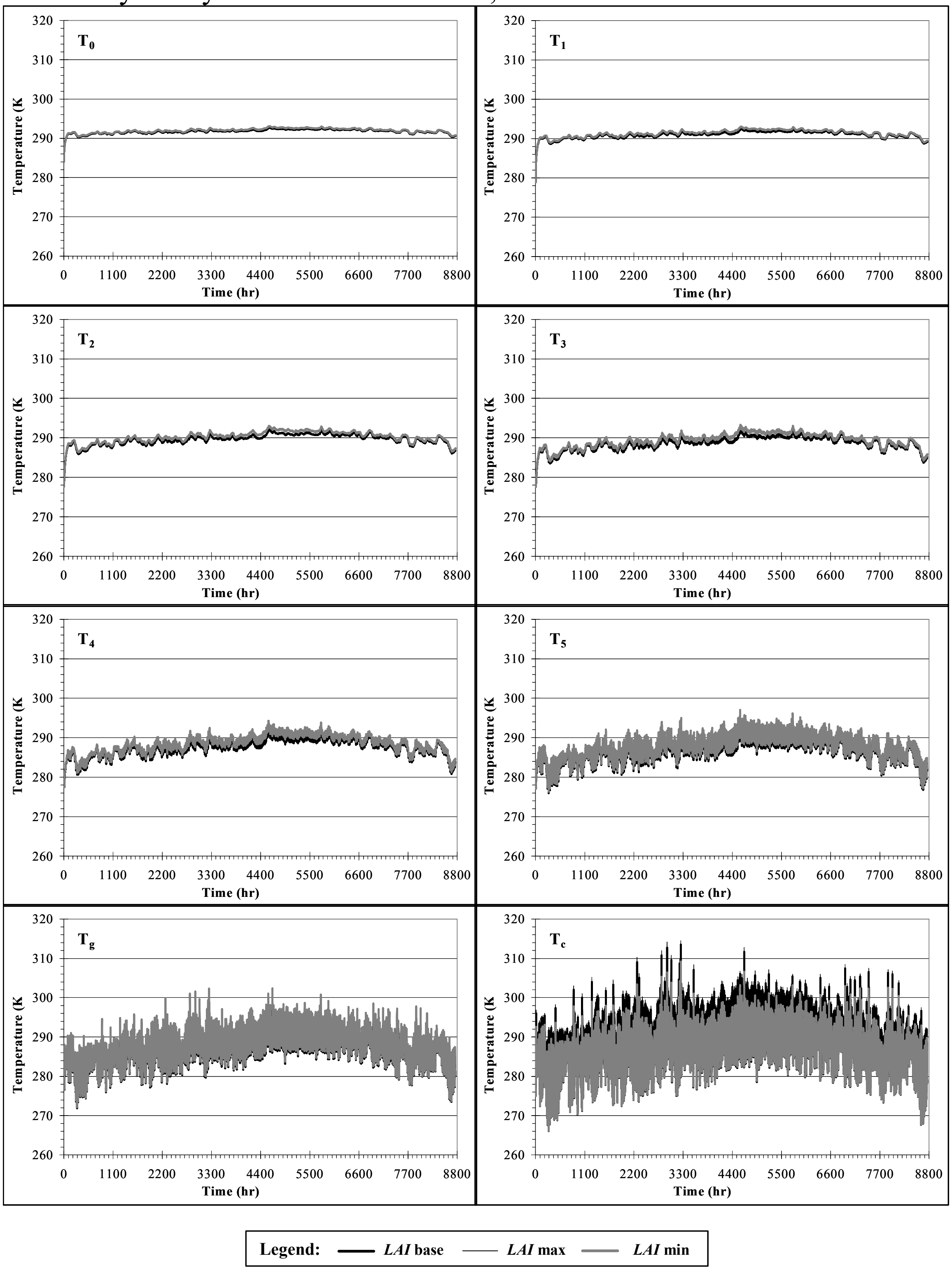
Sensitivity Analysis: k_{supp} – Santa Maria, CA



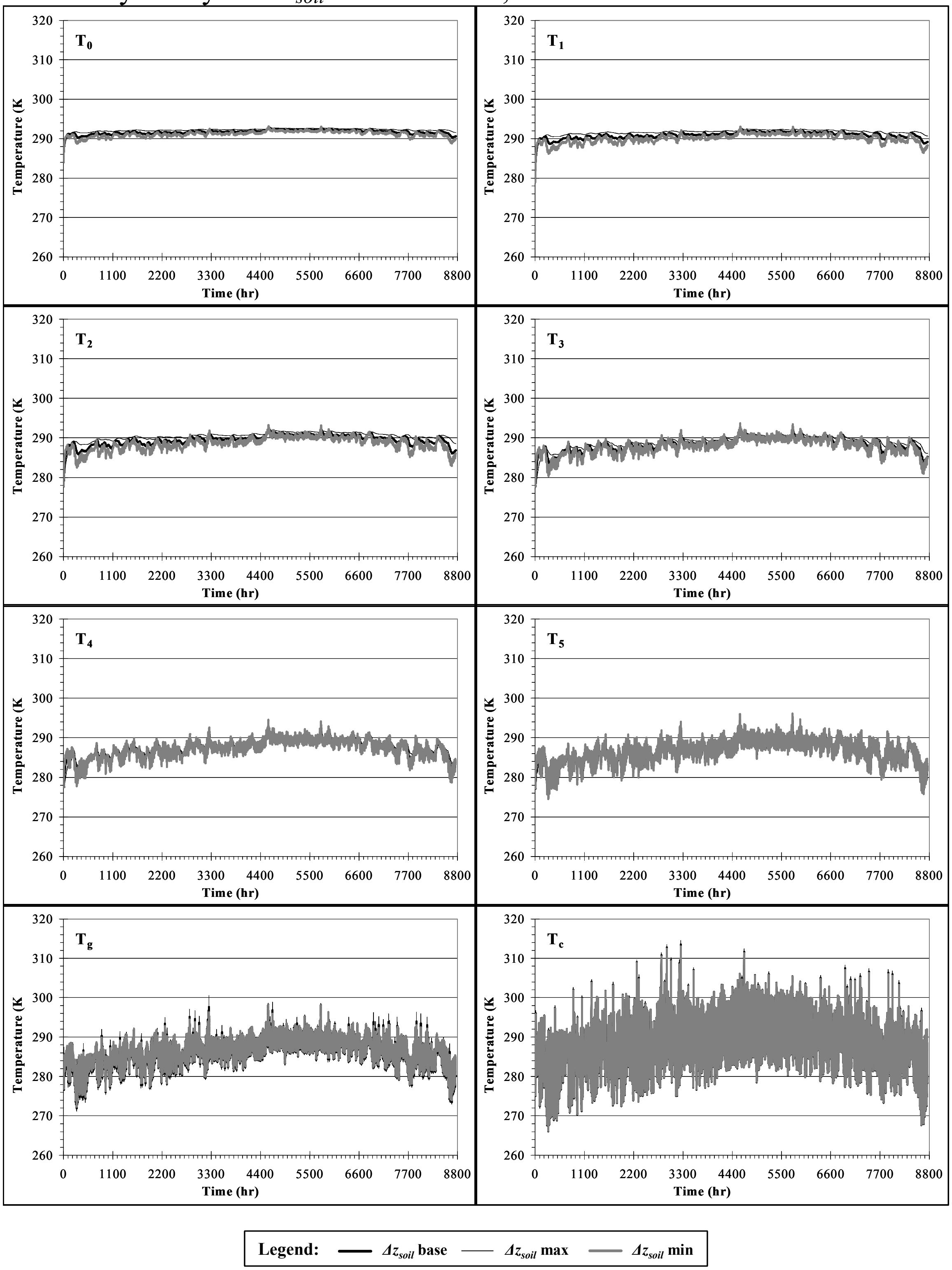
Sensitivity Analysis: ρ_g – Santa Maria, CA



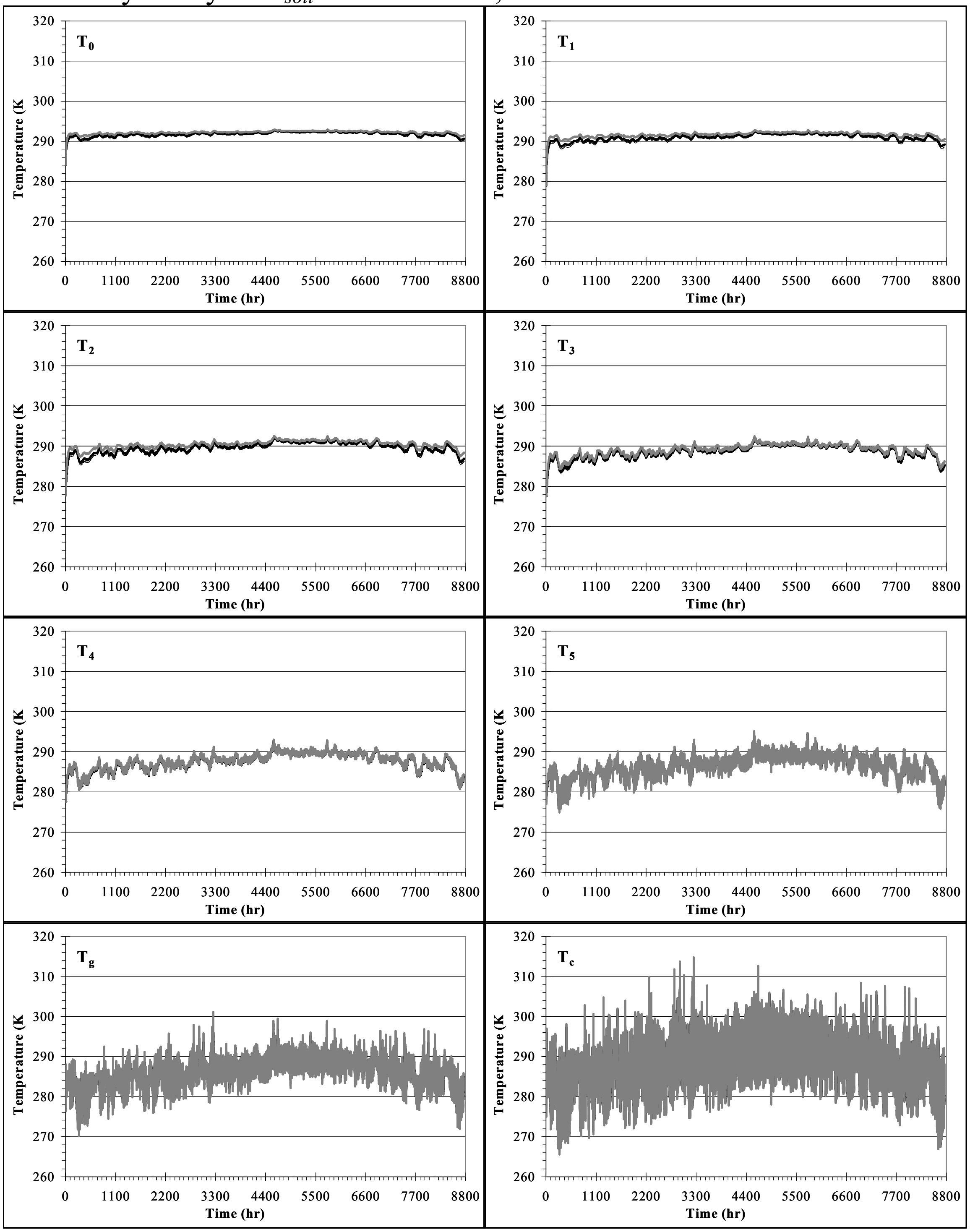
Sensitivity Analysis: *LAI* – Santa Maria, CA



Sensitivity Analysis: Δz_{soil} – Santa Maria, CA



Sensitivity Analysis: k_{soil} – Santa Maria, CA



Legend: k_{soil} base k_{soil} max k_{soil} min

Sensitivity Analysis: α_2 – Santa Maria, CA